Redundancy Allocation to Maximize a Lower Percentile of the System Time-to-Failure Distribution

David W. Coit, *Member IEEE*
Rutgers University, Piscataway

Alice E. Smith, *Member IEEE*
University of Pittsburgh, Pittsburgh

Key Words — System reliability, Reliability optimization, Redundancy allocation, Genetic algorithm.

Summary & Conclusions — An algorithm is presented which solves the redundancy-allocation problem when the objective is to maximize a lower percentile of the system time-to-failure distribution. The algorithm uses a genetic algorithm to search the prospective solution-space and a bisection search as a function evaluator. Previously, the problem has most often been formulated to maximize system reliability. For many engineering-design problems, this new formulation is more appropriate because there is often no clearly defined mission time on which to base component & system reliability. Additionally, most system designers & users are risk-averse, and maximization of a lower percentile of the system time-to-failure distribution is a more conservative (less risky) strategy compared to maximization of the mean or median time-to-failure. Results from over 60 examples clearly indicate that the preferred system design is sensitive to the user's perceived risk. We infer from these results that engineering-design decisions need to consider risk explicitly, and use of mean time-to-failure as a singular measure of product integrity is insufficient. Similarly, the use of system reliability as the principal performance measure is unwise unless mission time is clearly defined.

1. INTRODUCTION

Acronyms[1] *& Abbreviations*

GA	genetic algorithm
TTF	time-to-failure
NFT	near feasible threshold
pdf	probability density function
Cdf	cumulative distribution function

Notation[2]

$T(\mathbf{x}:\mathbf{k})$	system TTF for design solution $\mathbf{x}$, given $\mathbf{k}$
$T_{1-\alpha}(\mathbf{x}:\mathbf{k})$	α quantile of $T(\mathbf{x}:\mathbf{k})$ Cdf
α	system user's risk level: $0 < \alpha < 1$
$\lambda_{i,j}, \beta_{i,j}$	Weibull [pseudo-scale, shape] parameter for component j available to be used in subsystem i
$R(t, \mathbf{x}:\mathbf{k})$	system reliability (at time t) for design solution $\mathbf{x}$, given $\mathbf{k}$
$x_{i,j}$	quantity of available component j used in subsystem i; $x_{i,j} \in \{0, 1, 2, \dots\}$ unless otherwise specified
m_i	number of available components for subsystem i
s	number of subsystems
$\mathbf{x}_i$	$(x_{i,1}, x_{i,2}, \dots, x_{i,m_i})$
$\mathbf{x}$	$(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s)$
$\mathbf{x}^T$	transpose of $\mathbf{x}$
n_i	$\sum_{j=1}^{m_i} x_{i,j}$: number of components in subsystem i
$\mathbf{k}$	$(k_1, k_2, \dots, k_s)$
k_i	minimum number of components in parallel required for subsystem i to operate
NFT_i	NFT for constraint i
z	maximum $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$
$T_{i,j}$	TTF for component j available to be used in subsystem i
q	number of linear constraints
r	number of $x_{i,j}$ decision variables
$\mathbf{A}$	$q \times r$ matrix
$\mathbf{b}$	a q-dimension vector of the constraints

[1]The singular & plural of an acronym are always spelled the same.
[2]Other, standard notation is given in "Information for Readers & Authors" at the rear of each issue.

1.1 Background

Existing reliability-optimization algorithms are either inadequate or limited when the problem is to select a design configuration where there is no obvious choice for mission time, and thus, system reliability is not an appropriate performance measure. The redundancy-allocation problem involves selection of components and levels-of-redundancy to maximize some defined objective function. This paper formulates the problem to maximize an α lower quantile of the system TTF Cdf. A GA-based approach is used to solve the problem. This approach offers distinct benefits because it:

- does not require specification of a mission time,
- incorporates designer and system user risk.

0018-9529/98/$10.00 ©1998 IEEE

When system reliability is used as the primary performance measure, then there must be a defined mission time. For most consumer products, there is no one single time period of particular interest because the product is intended to be used for an extended, but ultimately unspecified, time.

Designers and product users are also generally risk-averse. If some tangible subset of the population fails very early, the product is viewed as unreliable by many consumers, even if the mean or median TTF is very high. This is particularly true if the implications of failure are severe. A conservative design-strategy is therefore to select the design which maximizes a $1-\alpha$ lower-bound on system TTF, *viz*, the α quantile of the system TTF Cdf. This provides better assurances that even the less reliable members of a production run are satisfactory.

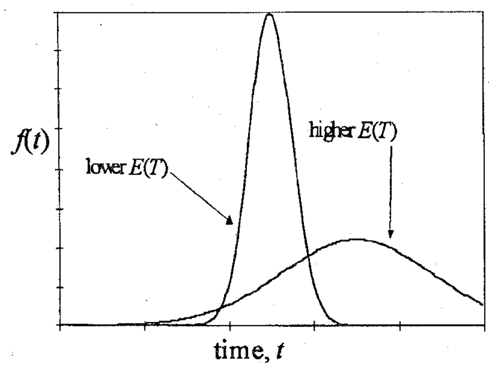

Figure 1: Comparison of System TTF Pdf

Consider the system TTF pdf in figure 1 for two functionally equivalent systems. A risk-neutral system-designer would prefer the system with the higher mean TTF, E[T]. For risk-averse designers, the choice is less clear. If the consequence of failure is very dire, then the choice in figure 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 ($<\alpha$). After this time period, it might be necessary to replace the system or to perform preventive maintenance. The α is user-selected and depends on the consequence of an early failure.

1.2 Assumptions

1. Components and the system are 2-state (good, bad).
2. Component TTF follows the Weibull distribution with known, deterministic parameters.
3. All redundancy is active (all components are operating).
4. There is no component (or system) repair or preventive maintenance[3].
5. Failures of individual components are mutually *s*-independent.
6. Failed components do not damage the system.
7. The system has s k-out-of-n:G subsystems in series.

[3]Many consumer items like toasters & radios are repairable, but consumer expectations are that repairs will not be needed for a very long time. The entire system is often discarded when a failure occurs.

2. REDUNDANCY[4] ALLOCATION PROBLEM

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

In this paper, the redundancy-allocation problem pertains to a system (from assumption #7). For each subsystem, there are m_i functionally equivalent components, with various levels of cost, weight, reliability, and other characteristics, which can be selected to be used in the system. There is an unlimited supply of each of the m_i choices. It is necessary to select a minimum of k_i components for each subsystem, but it might also be desirable to include additional components ($n_i > k_i$). There are system-level constraints; the problem is to select the design configuration that maximizes a stated objective function. The problem is NP-hard [1] and has been solved with system reliability as the objective function using dynamic programming [2, 3], integer programming [4, 5], and genetic algorithms [6 - 9].

System reliability is a mathematical convenient objective-function because (for a series system) it is the product of subsystem reliabilities, and its logarithm is a separable function. Dynamic programming or integer programming (with appropriate transformations) can then be used to determine optimal solutions to the problem. When there is no mission time to compute component & system reliability, these algorithms no longer apply.

If there is no obvious choice for mission time, design evaluation & optimization should be based on comparisons of the system TTF distributions (as opposed to the reliability for one distinct time period). It is common for many industries, including defense, railroad, and automotive, to use the mean TTF as a performance measure. It is less common to use a α quantile of the TTF Cdf; however, this can be more informative because it incorporates risk. It is generally insufficient to design & manufacture products which are highly reliable "on the average". It is necessary for a large fraction of the products to achieve some minimally acceptable performance level as well.

Painton & Campbell [7, 8] solved a formulation of the redundancy-allocation problem which incorporates risk. Component TTF has an exponential distribution, but the

[4]The terms, *series* & *parallel* are used in their logic-diagram sense, irrespective of the schematic-diagram or physical-layout.

distribution parameter itself is a r.v. They used a GA to maximize a lower quantile of the mean time-between-failures, given component-repair assumptions.

Ushakov & Harrison [10] and Gnedenko & Ushakov [11] present algorithms to maximize the median TTF (a surrogate for MTTF). Their heuristics can be applied if: 1) there is an appreciable amount of standby redundancy, and 2) $k_i = 1$ for every subsystem. Nakashima & Yamato [12] solve an analogous problem to maximize the time period where system reliability remains above a preselected value; their algorithm assumes that components have exponential TTF, and that the distribution parameters are the decision variables to be determined in addition to the redundancy levels.

2.1 Formulation to Maximize $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$

Problem P1 maximizes a quantile of the system TTF. The TTF for each available component has a 2-parameter Weibull distribution[5]. The α is pre-determined.

Problem P1 maximize $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$
subject to $\mathbf{A} \times \mathbf{x}^T \leq \mathbf{b}$
$k_i \leq \sum_{j=1}^{m_i} x_{i,j}$, for all i
where $T_{i,j}$ = TTF for a given (i,j)
$r_{i,j}(t) = \exp\left((-t/\theta_{i,j})^{\beta_{i,j}}\right) = \exp\left(-\lambda_{i,j} \cdot t^{\beta_{i,j}}\right)$
where $\lambda_{i,j} \equiv (1/\theta_{i,j})^{\beta_{i,j}}$

Typical constraints can include a cost-budget-limit or the maximum acceptable weight or volume.

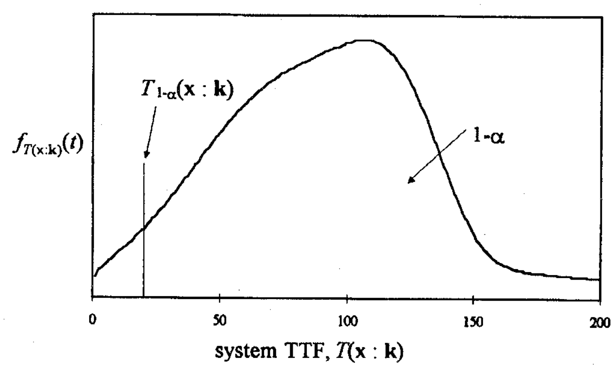

Figure 2: Example Pdf for $T(\mathbf{x}:\mathbf{k})$

Figure 2 depicts a typical pdf of the system TTF, $T(\mathbf{x}:\mathbf{k})$, and its lower percentile, $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$. The system pdf is complicated and non-standard for almost all cases, and depends upon the system structure-function and the component pdf's. The problem objective is to search over all feasible solutions, $\mathbf{x}$, to identify the design configuration which maximizes $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$. The optimal design configuration for a risk-neutral design is generally different than the optimal solution for a risk-averse design (*eg*, $\alpha = 0.05$).

A risk-neutral design objective involves maximizing the mean of a r.v., such as TTF, as described by Bunn [13]. A risk-neutral design objective can often be closely approximated by using the median TTF ($\alpha = 0.50$). Problem P1 is more realistic than other formulations for many actual design problems, but it has inherent difficulties. $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$ is intractable for any non-trivial problem. Usually, if a performance measure cannot be efficiently determined, then it is difficult to be included in an optimization algorithm. Monte Carlo simulation can be used to estimate $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$, similar to the approach of Painton & Campbell [7]; however, this can be inefficient, particularly during GA search where many prospective solutions must be evaluated.

To remedy this, the original problem can be restated using these definitions.

$$\begin{aligned} z &= \max\left[T_{1-\alpha}(\mathbf{x}:\mathbf{k})\right] \\ &= \max\left[t' : \Pr\{T(\mathbf{x}:\mathbf{k}) \geq t'\} \geq 1-\alpha\right] \\ &= \max\left[t' : R(t', \mathbf{x}:\mathbf{k}) \geq 1-\alpha\right] \end{aligned}$$

In this restated objective function, the t' is a continuous variable which can be considered a pseudo-mission time. The objective becomes to find the $\mathbf{x}$ associated with the largest pseudo-mission time which meets an additional constraint for system reliability at time t'. By adding reliability to the constraint set, the problem is equivalently restated as P2.

Problem P2 maximize t'
subject to $R(t', \mathbf{x}:\mathbf{k}) \geq 1-\alpha$
$\mathbf{A} \times \mathbf{x}^T \leq \mathbf{b}$
$k_i \leq \sum_{j=1}^{m_i} x_{i,j}$, for all i

An optimal solution to Problem P2 $(\mathbf{x}, t')$ consists of the optimal solution, $\mathbf{x}$, for Problem P1 and the optimal value of the objective function, $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$. The $R(t, \mathbf{x}:\mathbf{k})$ can be expressed as a function of the $x_{i,j}$, using:

$$R(t, \mathbf{x}:\mathbf{k}) = \prod_{i=1}^{s} R_i(t, \mathbf{x}:\mathbf{k}) \quad (1)$$

$$= \prod_{i=1}^{s}\left[1 - \sum_{l=0}^{k_i-1} \sum_{S_l} \prod_{j=1}^{m_i} binm\left(v_j; r_{i,j}(t), x_{i,j}\right)\right]$$

$$S_l \equiv \left\{\mathbf{v} : \left(\sum_{j=1}^{m_i} v_j\right) = l\right\}$$

$$\mathbf{v} \equiv (v_j : j = 1, \ldots, m_i)$$

$$r_{i,j}(t) \equiv \exp\left(-\lambda_{i,j} \cdot t^{\beta_{i,j}}\right)$$

v_j = number of unfailed (functioning properly) components j, $v_j \in \{0, 1, 2, \ldots, x_{i,j}\}$

Eq (1) accommodates the case where different, but functionally interchangeable, components are used in parallel; it involves the sum of s-independent joint binomial

[5]The θ is a true scale parameter, but is inconvenient to use in most calculations; the λ is a pseudo-scale parameter which is much more convenient.

events. Subsystem i has n_i components; see also assumption #5. For subsystem i, the probabilities are added, for all possible combinations of components which collectively result in less than k_i operational components. The first sum adds the probabilities of having l operational components less than k_i. The second sum adds the probabilities associated with all possible combinations, among the $(x_{i,j} : j = 1, \ldots, m_i)$, which sum to l operational components. For each combination, the product term is: Pr{'exactly v_1 of the $x_{i,1}$ are operational' AND 'v_2 of the $x_{i,2}$ components are operational', …, AND '$v_{m,i}$ of the x_i, m_i are operational'}.

$R(t, \mathbf{x} : \mathbf{k})$ is monotonically decreasing with time for any s-coherent system. Also, the original constraints do not include t'. Thus, for any prospective $\mathbf{x}$, the maximum t' can be found by increasing t' until: $R(t', \mathbf{x} : \mathbf{k}) = 1 - \alpha$ (the reliability constraint is tight).

3. SOLUTION ALGORITHM

TGA is a solution algorithm for this problem based on GA search, and expressed as TGA(α), which uses: 1) the GA originally defined in [9], together with a bisection search, to determine t', and 2) an adaptive penalty function to enforce compliance to the constraints.

TGA(α): Genetic Algorithm to Maximize $T_{1-\alpha}(\mathbf{x} : \mathbf{k})$

0. Initialize GA [9]. 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).
1. Determine Initial Population. Repeat the following items p times.
   · randomly select integers n_i ($\geq k_i$) 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.
2. Crossover Operation [9].
3. Mutation Operation [9].
4. Objective Function Determination - for each member of population.
   · Bisection search to determine t';
   · Adaptive penalty function [14].
5. Culling/Ranking Operation.
6. Termination Criteria Satisfaction.
   · If satisfied, select best feasible solution in final population as recommended design;
   · Else, proceed to next generation - go to Step 2;
   · End_If.

GA involves the evaluation of a population of solutions, which are revised over successive generations. Each solution is represented in the population by a vector.

The crossover & 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 which 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.

To determine the maximum t' for a particular $\mathbf{x}$, a bisection search is used to determine t' where $R(t', \mathbf{x} : \mathbf{k}) = 1 - \alpha$. Since system reliability is monotonically decreasing and there is generally some prior knowledge of the system TTF distribution, it is relatively easy through trial-and-error to find upper & lower bounds for t' to serve as starting points. These bounds are t_L & t_H:

$R(t_H, \mathbf{x} : \mathbf{k}) \leq 1 - \alpha \leq R(t_L, \mathbf{x} : \mathbf{k})$, $t_H = 10 t_L$.

After finding the appropriate starting points, the bisection search is used to determine t' to any pre-determined level of accuracy. The bisection 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 stopped when the upper & lower bound were within 0.001% of the maximum t'. Determination of the starting points generally required less than 5 function-evaluations, and then a maximum of 20 additional function-evaluations where required to estimate t' to within 0.001%. This is appreciably more efficient than using Monte Carlo simulation to estimate $T_{1-\alpha}(\mathbf{x} : \mathbf{k})$ to the same degree of accuracy. To make the search even more efficient, the bisection search can be replaced by a Fibonacci-series or golden-section search, or a derivative-based search such as Newton-Raphson.

The adaptive penalty function [14] was specifically developed to exploit information available from a GA search and to be updated by the relative success of the search as it proceeds. It promotes a thorough search within a NFT near the boundary between the feasible & infeasible regions. The penalized objective function then becomes: $t' - P(\mathbf{x})$.

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

$$\text{NFT}_i \equiv \frac{\text{NFT}_{i,o}}{1 + \Lambda_i \cdot g},$$

$\Delta b_i(\mathbf{x})$ = constraint violation for constraint i;
$\Delta b_i(\mathbf{x}) = 0$ if constraint is not violated;

Table 1: Input Parameters for Example 1

	subsystem 1				subsystem 3			
option	$\lambda_{i,j}$	$\beta_{i,j}$	$c_{i,j}$	$w_{i,j}$	$\lambda_{i,j}$	$\beta_{i,j}$	$c_{i,j}$	$w_{i,j}$
1	.0051	1.0	2	5	.00083	2.0	4	4
2	.0229	0.5	1	4	.0333	0.5	3	5
3	.0298	0.5	2	2	.0014	2.0	1	6
4	.000001	5.0	1	3	.0514	0.5	2	7

	subsystem 2				subsystem 4			
option	$\lambda_{i,j}$	$\beta_{i,j}$	$c_{i,j}$	$w_{i,j}$	$\lambda_{i,j}$	$\beta_{i,j}$	$c_{i,j}$	$w_{i,j}$
1	.0051	1.0	2	8	.044	0.5	4	6
2	.00062	2.0	1	10	.000002	5.0	5	4
3	.00073	2.0	1	9	.0019	2.0	3	5

Table 2: Algorithm Performance for Example 1 [10 TGA(α) runs]

α	avg # gen	max $T_{1-\alpha}(\mathbf{x})$	min $T_{1-\alpha}(\mathbf{x})$	avg $T_{1-\alpha}(\mathbf{x})$	# optimal	# feasible
.50	233.1	195.50	191.60	195.11	9/10	10/10
.10	372.5	46.58	46.58	46.58	10/10	10/10

$\text{NFT}_{i,o}, \Lambda_i =$ penalty-function constants;

$g =$ generation number for GA search, $g \in \{1, 2, \dots\}$;

$V_{\text{all}} =$ maximum unpenalized t' (feasible or infeasible);

$V_{\text{feas}} =$ maximum feasible t' found by the GA at that point in the search.

This $P(\mathbf{x})$ encourages the evaluation of infeasible solutions which are near the feasible region. It has been demonstrated using many reliability optimization sample problems [15] that a thorough GA search near the feasibility boundary, including both feasible & 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: 1) a pre-determined maximum number of generations, or 2) until no additional improvement is observed. 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. It is recommended that multiple TGA(α) runs be performed with various initial populations. Then, the best feasible solution should be used if all TGA(α) runs do not converge to the same solution.

4. ILLUSTRATIVE EXAMPLES[6]

The algorithm was demonstrated using:

Example 1 – a small illustrative example solved for α = 0.50 & 0.10;

Example 2 – modified versions of the 33 example-problems solved by Nakagawa & Miyazaki [3] for α = 0.50 & 0.05.

[6]The number of significant figures is not intended to imply any accuracy in the estimates, but to illustrate the arithmetic.

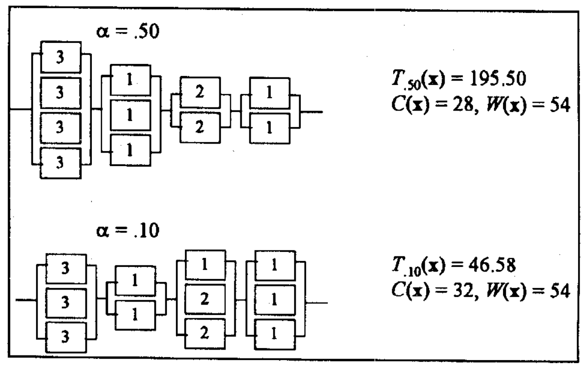

Figure 3: Problem Solutions for Example 1

For both examples, k_i = 1, for all i; a GA population size of 40 was used; 10 different TGA(α) runs were made with different initial populations for each test case.

4.1 Example 1

Example 1 input parameters are presented in table 1. The system-level cost and weight constraints are 32 and 54, respectively. The $c_{i,j}$ & $w_{i,j}$ are the cost & weight of available component j for subsystem i. When α = 0.50, the problem closely approximates a risk-neutral design situation and involves the maximization of the median TTF. When α = 0.10, the problem is for a risk-averse design situation.

Example 1 results are presented in figure 3 and table 2. The selection of components among the subsystems varies depending on α, as figure 3 shows. For both optimal

Table 3: Input Parameters for Example 2

	choice 1			choice 2			choice 3			choice 4		
i	$\lambda_{i,j}$	$\beta_{i,j}$	E[T]	$\lambda_{i,j}$	$\beta_{i,j}$	E[T]	$\lambda_{i,j}$	$\beta_{i,j}$	E[T]	$\lambda_{i,j}$	$\beta_{i,j}$	E[T]
1	.0051293	1.0	195	.0229489	0.5	3798	.0298237	0.5	2249	.0000011	5.0	14
2	.0051293	1.0	195	.0006188	2.0	36	.0007257	2.0	33	—	—	—
3	.0008338	2.0	31	.0333179	0.5	1802	.0013926	2.0	24	.0513930	0.5	757
4	.0440385	0.5	1031	.0000016	5.0	13	.0018633	2.0	21	—	—	—
5	.0051293	1.0	195	.0195667	0.5	5224	.0007257	2.0	33	—	—	—
6	.0010050	1.0	995	.0002020	2.0	62	.0000003	5.0	19	.0000004	5.0	17
7	.0000006	5.0	16	.0008338	2.0	31	.0298237	0.5	2249	—	—	—
8	.0000009	5.0	15	.0000011	5.0	14	.0000021	5.0	13	—	—	—
9	.0010050	1.0	995	.0030459	1.0	328	.0040822	1.0	245	.0000009	5.0	15
10	.0105361	1.0	95	.0513930	0.5	757	.0186330	1.0	54	—	—	—
11	.0040822	1.0	245	.0005129	2.0	39	.0061875	1.0	162	—	—	—
12	.0010536	2.0	27	.0162519	1.0	62	.0198451	1.0	50	.0023572	2.0	18
13	.0001005	2.0	88	.0000002	5.0	20	.0030459	1.0	328	—	—	—
14	.0001005	2.0	88	.0000005	5.0	17	.0000008	5.0	15	.0000011	5.0	14

designs, 11 components are used, but the distribution of the 11 components among the 4 subsystems depends on the risk level. Table 2 presents: avg # gen – average number of generations before the algorithm converged to a solution,
max $T_{1-\alpha}(\mathbf{x})$ – maximum objective functions,
min $T_{1-\alpha}(\mathbf{x})$ – minimum objective functions,
avg $T_{1-\alpha}(\mathbf{x})$ – average objective functions,
opt – number of final solutions (out of 10) which were optimal,
feas – number of final solutions (out of 10) which were feasible.

4.2 Example 2

In example 2, the system has 14 subsystems with 3 or 4 component choices 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 [3].

For this problem, Weibull parameters were selected for each component to be consistent with the reliability values specified in the original problem statement [2]. To generate a set of component Weibull parameters ($\lambda_{i,j}$, $\beta_{i,j}$), we assumed that the previously specified reliability values were for a mission time of 10 time units.

· $\beta_{i,j}$ values were randomly & uniformly selected from among 0.5, 1, 2, 5 — ranging from infant mortality to severe wear-out.

· $\lambda_{i,j}$ values were computed based on $\beta_{i,j}$ and the previously specified reliability at 10 time units.

Table 3 shows the resulting input parameters of the available components for the 14 subsystems. The input parameters are completely consistent with those originally specified if the objective is to maximize system reliability for a mission time of 10. The values for $c_{i,j}$, $w_{i,j}$ were previously presented [2,3].

Figure 4 presents reliability *vs* time for the 3 component-choices available for subsystem 4. At time 10, the component reliability values are 0.87, 0.85, 0.83, as originally specified. Figure 4 shows the inherent danger associated with the use of algorithms which maximize system reliability if mission time is not clearly defined. Since TTF distribution information was not originally provided, the assigned $\beta_{i,j}$ & $\lambda_{i,j}$ values, are plausible and completely consistent with the original formulation. However, each of the 3 component-options are both the most reliable and least reliable for different time periods.

Example 2 results are presented in tables 4 & 5. 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 appreciably. This strongly implies that there needs to be different design strategies depending on a user's risk-profile.

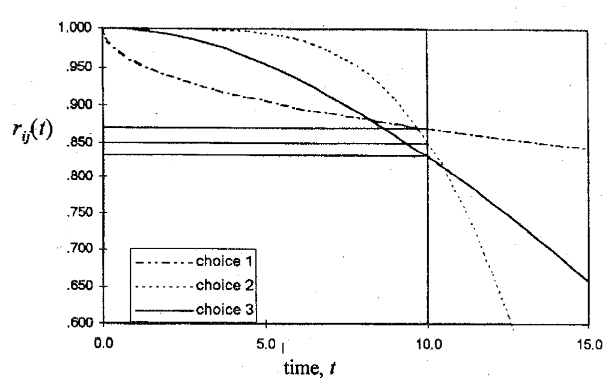

Figure 4: Example 2 (Subsystem 4) Component Reliability *vs* Time

Table 4: Example-2 Results ($\alpha = 0.5$)

TGA(α) Results – 10 runs					Max $T_{1-\alpha}(\mathbf{x}:\ \mathbf{k})$ Solution		
no.	max	min	mean	std dev	cost	weight	component selections
1	18.67	18.65	18.66	.0069	130	191	33,11,22,111,22,12,33,1111111,1,222,33,333,11,1
2	18.66	18.63	18.66	.0085	130	190	33,11,22,111,22,22,33,1111111,1,222,33,333,11,1
3	18.65	18.61	18.63	.0171	129	189	33,11,22,11,22,22,33,1111111,1,222,33,3333,11,1
4	18.63	18.62	18.63	.0047	128	188	33,11,22,111,22,1,33,1111111,1,222,33,233,11,1
5	18.62	18.61	18.62	.0071	128	187	33,11,22,11,22,12,33,1111111,1,222,33,223,11,1
6	18.62	18.56	18.60	.0190	128	186	33,11,22,11,22,22,33,1111111,1,222,33,223,11,1
7	18.61	18.58	18.60	.0101	127	185	33,11,22,11,22,22,33,1111111,1,222,33,233,11,1
8	18.59	18.53	18.58	.0218	126	184	33,11,22,11,22,22,33,1111111,1,222,33,333,11,1
9	18.57	18.51	18.56	.0187	125	183	3,11,22,11,22,22,33,1111111,1,222,33,233,11,1
10	18.57	18.51	18.55	.0224	124	182	33,11,22,11,22,1,33,1111111,1,222,33,233,11,1
11	18.56	18.51	18.53	.0139	123	181	33,11,22,11,22,1,33,1111111,1,222,33,333,11,1
12	18.53	18.48	18.51	.0204	122	180	3,11,22,11,22,1,33,1111111,1,222,33,233,11,1
13	18.52	18.49	18.51	.0098	121	179	3,11,22,11,22,1,33,1111111,1,222,33,333,11,1
14	18.49	18.44	18.47	.0206	121	178	33,11,22,11,22,1,33,1111111,1,12,33,233,11,1
15	18.48	18.44	18.47	.0125	120	177	33,11,22,11,22,1,33,1111111,1,22,33,233,11,1
16	18.47	18.44	18.46	.0135	119	176	33,11,22,11,22,1,33,1111111,1,22,33,333,11,1
17	18.44	18.40	18.43	.0186	118	175	3,11,22,11,22,1,33,1111111,1,22,33,233,11,1
18	18.43	18.37	18.41	.0252	117	174	3,11,22,11,22,1,33,1111111,1,22,33,333,11,1
19	18.40	18.36	18.39	.0149	117	173	3,11,22,11,23,1,33,1111111,1,22,33,333,11,1
20	18.39	18.36	18.37	.0133	117	172	33,11,22,11,22,1,33,1111111,1,22,33,233,1,1
21	18.37	18.35	18.36	.0139	116	171	33,11,22,11,22,1,33,1111111,1,22,33,333,1,1
22	18.35	18.30	18.34	.0196	115	170	3,11,22,11,22,1,33,1111111,1,12,33,333,1,1
23	18.34	18.23	18.28	.0512	114	169	3,11,22,11,22,1,33,1111111,1,22,33,333,1,1
24	18.31	18.15	18.28	.0523	114	168	3,11,22,11,23,1,33,1111111,1,22,33,333,1,1
25	18.26	18.14	18.24	.0410	115	167	3,11,12,11,23,1,33,1111111,1,22,33,333,1,1
26	18.22	18.11	18.18	.0303	113	166	3,11,22,11,22,1,33,1111111,1,22,33,22,1,1
27	18.18	18.09	18.12	.0331	113	165	3,11,22,11,23,1,33,1111111,1,22,33,22,1,1
28	18.14	18.05	18.10	.0326	112	164	3,11,22,11,23,1,33,1111111,1,22,33,23,1,1
29	18.11	18.08	18.09	.0127	114	163	33,1,22,11,22,1,33,1111111,1,22,33,333,1,1
30	18.09	18.04	18.05	.0177	113	162	3,1,22,11,22,1,33,1111111,1,12,33,333,1,1
31	18.07	17.89	18.01	.0562	112	161	3,1,22,11,22,1,33,1111111,1,22,33,333,1,1
32	18.04	17.89	17.97	.0591	112	160	3,1,22,11,23,1,33,1111111,1,22,33,333,1,1
33	17.99	17.84	17.91	.0677	113	159	3,1,12,11,23,1,33,1111111,1,22,33,333,1,1

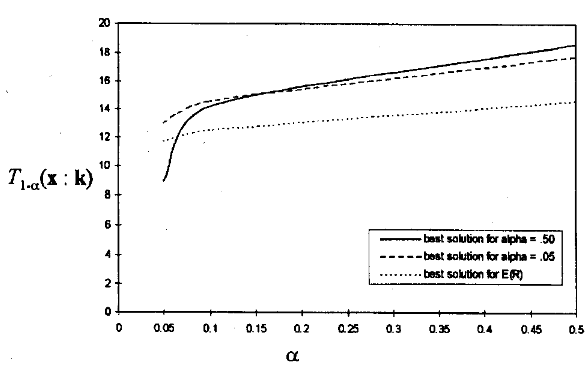

Figure 5: Comparison of Best Solutions (Example 2, #1)

This observation is further demonstrated by the results in figure 5, for the least constrained case (weight-constraint = 191). Figure 5 presents $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$ as a function of α for 3 design configurations; the solution which maximizes system reliability (at time 10), and the TGA(α) solutions for $\alpha = 0.50$ and 0.05 respectively. Figure 5 shows that no design displays first-order stochastic dominance [13].

The solution which maximizes reliability is not particularly effective at maximizing $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$ at any α-level. These results further support the idea that maximization of reliability might not be an effective design strategy. The solution which maximizes $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$ for α=0.50 is a poor choice for maximizing $T_{1-\alpha}(\mathbf{x}:\mathbf{k})$ at α=0.05; similarly, the reverse argument is also true.

The results in tables 4 & 5 indicate that TGA(α) performance was very consistent, although each run did not always converge to the same solution. All 66 test cases had a standard deviation (of the 10 TGA(α) runs) less than 1% of the mean objective-function value.

Table 5: Example-2 Results ($\alpha = 0.05$)

TGA(α) Results – 10 runs					Max $T_{1-\alpha}(\mathbf{x}:\ \mathbf{k})$ Solution		
no.	max	min	mean	std dev	cost	weight	component selections
1	13.00	12.84	12.97	.0640	130	191	33,11,111,111,333,22,33,1111,23,222,33,3334,11,12
2	12.91	12.74	12.87	.0560	129	190	33,11,111,111,333,22,33,1111,23,222,33,3344,11,12
3	12.82	12.75	12.78	.0180	130	189	33,11,111,111,333,22,33,1111,33,222,33,3344,12,12
4	12.74	12.68	12.72	.0320	130	188	33,11,111,111,333,22,33,1111,33,222,33,3444,11,12
5	12.62	12.52	12.58	.0390	128	187	33,11,111,111,333,22,33,1113,33,222,33,3344,11,12
6	12.52	12.28	12.45	.0790	127	186	33,11,111,111,333,22,33,1113,33,222,33,3444,11,12
7	12.40	12.27	12.36	.0480	126	185	33,11,111,113,333,22,33,1113,33,222,33,3444,11,12
8	12.29	12.17	12.27	.0400	126	184	33,11,111,111,333,22,33,111,33,222,33,3334,11,12
9	12.24	12.02	12.16	.0720	125	183	33,11,111,111,333,22,33,111,33,222,33,3344,11,12
10	12.16	11.92	12.07	.0780	124	182	33,11,111,111,333,22,33,111,33,222,33,3444,11,12
11	12.07	11.92	12.01	.0580	123	181	33,11,111,113,333,22,33,111,33,222,33,3444,11,12
12	11.97	11.81	11.92	.0470	122	180	33,11,111,113,333,22,33,111,33,222,33,4444,11,12
13	11.88	11.72	11.83	.0500	124	179	33,11,111,112,333,22,33,111,33,222,33,4444,11,12
14	11.73	11.51	11.66	.0680	125	178	33,11,111,112,333,22,33,111,33,222,33,333,11,12
15	11.61	11.53	11.57	.0270	122	177	33,11,111,112,23,22,33,111,33,222,33,4444,11,12
16	11.47	11.35	11.42	.0340	122	176	33,11,111,112,33,22,33,111,33,222,33,4444,11,12
17	11.37	11.17	11.29	.0530	122	175	33,11,111,122,333,22,33,113,33,222,33,4444,11,12
18	11.36	11.17	11.23	.0550	119	174	33,11,111,112,333,22,33,111,1,222,33,4444,11,12
19	11.17	11.04	11.14	.0410	119	173	33,11,111,123,333,22,33,113,33,222,33,4444,11,22
20	11.18	11.00	11.04	.0620	123	172	33,11,111,2222,333,22,33,113,1,222,33,4444,11,12
21	11.02	10.92	10.99	.0290	124	171	333,11,11,2222,333,22,33,113,33,122,33,4444,11,22
22	10.91	10.84	10.88	.0250	124	170	33,11,111,2222,333,22,33,113,33,222,33,144,11,1
23	10.84	10.76	10.82	.0280	124	169	33,11,111,2222,333,22,33,113,1,222,33,134,11,22
24	10.80	10.71	10.76	.0270	117	168	33,11,111,122,333,22,33,113,1,122,33,4444,11,22
25	10.75	10.63	10.69	.0430	116	167	33,11,111,122,333,22,33,113,1,222,33,4444,11,22
26	10.67	10.58	10.62	.0280	115	166	33,11,11,222,333,22,33,3333,33,222,33,4444,11,22
27	10.63	10.45	10.56	.0480	117	165	33,11,111,222,333,22,33,113,1,222,33,4444,11,22
28	10.55	10.47	10.50	.0290	115	164	33,11,11,222,333,22,33,133,33,222,33,4444,11,22
29	10.47	10.36	10.43	.0330	116	163	33,11,11,222,33,22,33,113,33,222,33,4444,11,22
30	10.40	10.32	10.36	.0230	114	162	33,11,11,222,33,22,33,133,33,122,33,4444,11,22
31	10.36	10.26	10.30	.0280	113	161	33,11,11,222,33,22,33,133,33,222,33,4444,11,22
32	10.26	10.11	10.21	.0530	111	160	33,11,11,222,333,22,33,133,1,122,33,4444,11,22
33	10.21	10.18	10.16	.0190	110	159	33,11,11,222,333,22,33,133,1,222,33,4444,11,22

4.3 General Observations

The optimal solution was found for example 1 by enumerating the search space. For example 2, the optimal solutions are unknown. There are no competing algorithms to yield optimal results, and the search space is too large to enumerate. Example 2 demonstrates how the algorithm works, evaluates the consistency of its performance, and compares the resulting design strategies for various α.

TGA(α) can be generalized to consider a broader range of problems. Any parametric-component TTF distribution can be incorporated into it; and the penalty function can accommodate any form of non-linear constraints.

REFERENCES

[1] M.S. Chern, "On the computational complexity of reliability redundancy allocation in a series system", *Operations Research Letters*, vol 11, 1992 Jun, pp 309-315.

[2] D.E. Fyffe, W.W. Hines, N.K. Lee, "System reliability allocation and a computational algorithm", *IEEE Trans. Reliability*, vol R-17, 1968 Jun, pp 64-69.

[3] Y. Nakagawa, S. Miyazaki, "Surrogate constraints algorithm for reliability optimization problems with two constraints", *IEEE Trans. Reliability*, vol R-30, 1981 Jun, pp 175-180.

[4] R.L. Bulfin, C.Y. Liu, "Optimal allocation of redundant components for large systems", *IEEE Trans. Reliability*, vol R-34, 1985 Aug, pp 241-247.

[5] M. Gen, K. Ida, J.U. Lee, "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, vol 73, 1990 Mar, pp 88-96.

[6] K. Ida, M. Gen, T. Yokota, "System reliability optimization with several failure modes by genetic algorithm", *Proc. 16th Int'l Conf. Computers and Industrial Eng'g*, 1994 Mar, pp 349-352.

[7] L.A. Painton, J.E. Campbell, "Genetic algorithms in optimization of system reliability", *IEEE Trans. Reliability*, vol 44, 1995 Jun, pp 172-178.

[8] J.E. Campbell, L.A. Painton, "Optimization of reliability allocation strategies through use of genetic algorithms", *Proc. 6th Symp. Multidisciplinary Design and Optimization*, 1996 Sep, pp 1233-1242; AIAA/NASA/ISSMO.

[9] D.W. Coit, A.E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm", *IEEE Trans. Reliability*, vol 45, 1996 Jun, pp 254-260.

[10] I. Ushakov, R. Harrison, *Handbook of Reliability Engineering*, 1994, pp 513-515; John Wiley & Sons.

[11] B. Gnedenko, I. Ushakov, *Probabilistic Reliability Engineering*, 1995, pp 505-511; John Wiley & Sons.

[12] K. Nakashima, K. Yamato, "Optimal design of a series-parallel system with time-dependent reliability", *IEEE Trans. Reliability*, vol 26, 1977 Jun, pp 119-120.

[13] D. Bunn, *Applied Decision Analysis*, 1984; McGraw-Hill.

[14] D.W. Coit, A.E. Smith, D.M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems", *INFORMS J. Computing*, vol 8, 1996 Spring, pp 173-182.

[15] D.W. Coit, A.E. Smith, "Penalty guided genetic search for reliability design optimization", *Computers and Industrial Eng'g*, vol 30, 1996 Sep, pp 895-904.

AUTHORS

Dr. David W. Coit; Dep't of Industrial Eng'g; Rutgers Univ; 96 Frelinghuysen Road; Piscataway, New Jersey 08854-8018 USA.
Internet (e-mail): coit@rci.rutgers.edu

David W. Coit (Member IEEE) is an Assistant Professor at Rutgers University. He received a BS (1980) in Mechanical Engineering from Cornell University, an MBA (1988) from Rensselaer Polytechnic Institute, and MS & PhD (1993, 1996) in Industrial Engineering from the University of Pittsburgh. From 1980 to 1992, he was employed at IIT Research Institute (IITRI), Rome, NY. From 1980 to 1988, he was a reliability engineer and project manager at IITRI, where he developed reliability prediction and optimization models, and analyzed the reliability of engineering designs for client organizations. From 1988 to 1992, he was the Engineering Manager at the IITRI Assurance Technology Center. His current research involves reliability prediction and optimization, and industrial applications for artificial neural networks. He is a member of IEEE, IIE, and INFORMS.

Dr. Alice E. Smith; 1031 Benedum Hall; Dep't of Industrial Eng'g; Univ. of Pittsburgh; Pittsburgh, Pennsylvania 15261 USA.
Internet (e-mail): aesmith@engrng.pitt.edu

Alice E. Smith (Member IEEE) is Associate Professor of Industrial Engineering. After 10 years of industrial experience with Southwestern Bell Corporation, she joined the faculty of the University of Pittsburgh in 1991. Her research interests are in modeling & optimization of complex systems using computational intelligence techniques; her research has been sponsored by Lockheed Martin Corporation, the Ben Franklin Technology Center of Western Pennsylvania, and the US NSF, from which she was awarded a CAREER grant in 1995. She is an Associate Editor of *INFORMS Journal on Computing and Engineering Design and Automation*, and is a registered Professional Engineer in the state of Pennsylvania. Dr. Smith is a member of IEEE, ASEE, and INFORMS, and a Senior Member of IIE and SWE.

Manuscript TR97-029 received 1997 March 10;
revised 1998 January 27.
Responsible editor: R.J. Loomis Jr.
Publisher Item Identifier S 0018-9529(98)05609-7