Considering Risk Profiles in Design Optimization for Series-Parallel Systems

David W. Coit • Rutgers University • Piscataway

Alice E. Smith • University of Pittsburgh • Pittsburgh

Key Words: Design optimization, Genetic algorithms, Redundancy allocation

SUMMARY & CONCLUSIONS

The objective of the redundancy allocation problem is to select the parts to be used and a design configuration to optimize some objective function without violating system-level constraints. In this paper, the problem was formulated in a more general framework than has previously been accomplished, allowing for the explicit consideration of the risk-profile of system designers and users. Risk is related to the probability that a randomly selected system will be less reliable than a predicted or specified value. Previous formulations of the problem involved a maximization of the expected value of system reliability or the mean-time-to-failure, which implies that designers are risk-neutral. In this paper, both component reliability and time-to-failure are considered as random variables, and lower-bound limits are used in the optimization problem. The extent of the risk associated with a particular system design project was included in the problem formulations and influenced the final design decisions. These formulations were intended to more faithfully address the actual concerns and considerations of the engineering design community. The problems were then solved using a Genetic Algorithms (GAs). Several illustrative examples are provided which show different design configurations depending on the risk profile.

1. *INTRODUCTION*

Determination of an optimal or near optimal system design is very important to economically produce new systems which meet and exceed customers' expectations for reliability, performance, etc. When developing a new system, there are detailed engineering specifications which prescribe minimum levels of reliability, maximum weight, maximum volume, etc. The redundancy allocation problem, addressed in this paper, involves the simultaneous evaluation and selection of available components and a system-level design configuration which collectively meets all design constraints, and at the same time, optimizes some objective function, usually system cost or reliability. In practice, each of these parameters may not be known exactly and there is some element of risk that the constraint will not actually be met or the objective function value may not actually be achieved.

Prior formulations of the redundancy allocation problem have generally required deterministic reliability values to be known for all available component choices. Sophisticated optimization algorithms have then been developed to solve these problems but they have not always been practical for several reasons. One reason is that the use of average or expected reliability values within the constraints or objective function implies that the system designers and users are risk-neutral, when in fact, that is rarely the case. If the system reliability is a random variable (because all component reliability values are not known precisely) and there is a reliability constraint, system users desire a design which exceeds the requirement consistently, not simply "on average" as a risk-neutral approach might imply.

1.1 *Notation*

C_o	cost constraint
W_o	weight constraint
s	number of subsystems
m_i	quantity of available component choices for subsystem i ($i = 1,...,s$)
λ_{ij}	Weibull scale parameter of component j available for subsystem i
β_{ij}	Weibull shape parameter of component j available for subsystem i
$\mathbf{x}$	solution vector which defines the system design $= (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s)$
$\mathbf{x}_i$	solution vector for subsystem i
$\mathbf{x}_i$	$= (x_{i1}, x_{i2}, \ldots, x_{i,m_i})$
x_{ij}	quantity of the j^{th} component used in subsystem i
n_i	total number of components used in subsystem i $= x_{i1} + x_{i2} + \ldots + x_{i,m_i}$
$R(t_o, \mathbf{x})$	system reliability
$R_i(t_o, \mathbf{x}_i)$	reliability of subsystem i
$R_{1-\alpha}(t_o, \mathbf{x})$	$\alpha \times 100\%$ percentile of system reliability distribution
$C_i(\mathbf{x}_i)$	total cost of subsystem i
$W_i(\mathbf{x}_i)$	total weight of subsystem i
$T(\mathbf{x})$	system time-to-failure
$T_{1-\alpha}(\mathbf{x})$	$\alpha \times 100\%$ percentile of system time-to-failure distribution

α	risk level ($0 < \alpha < 1$)
t_o	mission time

1.2 *List of Assumptions*

1. Redundancy is active redundancy.
2. There is no component (or system) repair or preventive maintenance.
3. There are discrete component choices available for each required system function.
4. Each component used in a system is *s*-independent.
5. The system operates until failure.

2. *REDUNDANCY ALLOCATION PROBLEM*

Design and development of new products involves the selection of components and a system-level design configuration to satisfy detailed functional and performance specifications. For this problem, the overall system is partitioned into a specific number of subsystems in series. For each subsystem, there are multiple component choices which can be selected and used in parallel. Figure 1 presents a typical example of a series-parallel system.

For each subsystem (i=1,...,s), there are m_i discrete component choices available with associated reliability, cost, weight and/or other characteristics. These parameters may either be a constant or they may be described by a probability density function. For each subsystem, a minimum of one component must be chosen from among the m_i available choices (with an unlimited supply available for each of the m_i components). It may then become advisable, or even necessary, to place additional components in parallel to maximize the reliability. Additionally, there is the option of using two (or more) of a lower reliability component as an alternative to using a more reliable, but also more costly, component. The consumer electronics industry is one such example where entirely new system designs are often composed largely of standard component types (e.g., microcircuits, transistors, resistors) with known characteristics.

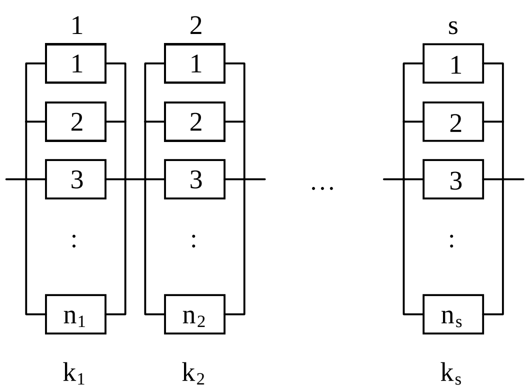

Figure 1: Series-Parallel System Configuration.

Previous formulations of the problem generally did not allow for variability in component characteristics. It has been assumed that all component-level information was constant and known precisely. Therefore, it was necessary to base the algorithm on component and system reliability, as opposed to time-to-failure, which is always considered a random variable. Also, the expected value or an assumed constant value for reliability, weight and/or cost had to be used (as opposed to a value described by a probability density function) within the objective functions and constraints. If component reliability values are truly constant, then the reliability over a stated mission time is known exactly and there is no risk that the population of systems has a lower reliability level.

However, this is neither an accurate representation of most engineering problems nor a particularly useful problem formulation for risk-averse system designers and users. There actually is variability present for real-world problems, and knowledge of a distribution for component reliability or time-to-failure are necessary for designers with different risk profiles.

A general problem formulation where the user's risk profile is considered and where component reliability values are stochastic is as follows. In the formulation, linear system-level constraints on cost and weight are shown as typical examples of problem constraints.

Problem P1: max $R_{1-\alpha}(t_o,\mathbf{x})$ (or $T_{1-\alpha}(\mathbf{x})$)

subject to
$$\sum_{i=1}^{s} C_i(\mathbf{x}_i) \le C$$
$$\sum_{i=1}^{s} W_i(\mathbf{x}_i) \le W$$
$$x_{ij} \in (0, 1, 2, ...)$$

$R_{1-\alpha}(t_o,\mathbf{x})$ is defined as the reliability level at which a randomly selected system will exceed that value with a probability equal to 1-α, or as follows.

$$R_{1-\alpha}(t_o,\mathbf{x}) = \left\{ r \,\middle|\, \Pr\bigl(R(t_o,\mathbf{x}) \ge r\bigr) = 1-\alpha \right\}$$

$T_{1-\alpha}(t_o,\mathbf{x})$ is defined similarly. The choice of whether to maximize $R_{1-\alpha}(t_o,\mathbf{x})$ or $T_{1-\alpha}(\mathbf{x})$ depends on whether there is a single mission time, t_o, of interest. Alternatively, the objective also could have been to minimize system cost and $R_{1-\alpha}(t_o,\mathbf{x})$ or $T_{1-\alpha}(\mathbf{x})$ could have been in a constraint with a minimum acceptable level specified. For either case ($R_{1-\alpha}(t_o,\mathbf{x})$ or $T_{1-\alpha}(\mathbf{x})$), the user's risk profile is considered in the problem by the α value, defined as the risk level, which represents the probability that a randomly chosen system will have a reliability value less than $R_{1-\alpha}(t_o,\mathbf{x})$ or that the time-to-failure will be less than $T_{1-\alpha}(\mathbf{x})$.

When $R_{1-\alpha}(t_o,\mathbf{x})$ is included in the problem formulation, it implies that component reliability values are not known precisely and system reliability is itself a random variable. This can occur if there is uncertainty in the time-to-failure distribution parameters, the reliability is determined empirically, or the mission time is not constant but also characterized by a distribution. Figure 2 shows the concept of a (1-α)×100% lower bound on reliability.

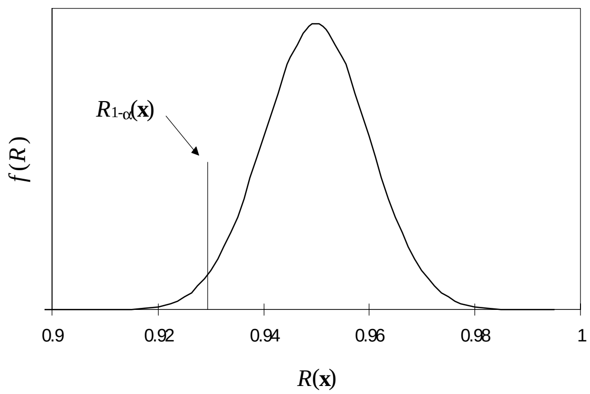

Figure 2: (1-α)×100% Lower-bound on Reliability.

For risk-averse design problems where the implications of failure are severe, it is not sufficient to design a system where the mean or median time-to-failure is exemplary. It is necessary to know that a large majority of items from the population will also achieve some satisfactory level. For a risk-neutral design scenario, an α value of 0.50 is appropriate, however, for a risk-averse design scenario, a low α value (e.g., $\alpha = .10$ or $.01$) should be used.

For problems without a clearly defined mission time, consider the two system time-to-failure probability density functions given in Figure 3. Each one represents a different system with everything equal except for reliability. The risk-neutral designer would naturally prefer the option with the higher MTTF. Alternatively, a risk-averse designer may not prefer this choice at all. If the implications of a failure are severe and the probability of a failure must be tightly controlled ($\alpha < .05$), the option with the lower MTTF is a better choice. This user can safely use this system for 3.5 time units without any appreciable prospect of a failure. At that time, it may then be necessary to replace the system with a new one, perform preventive maintenance or use it in a different, less critical, application.

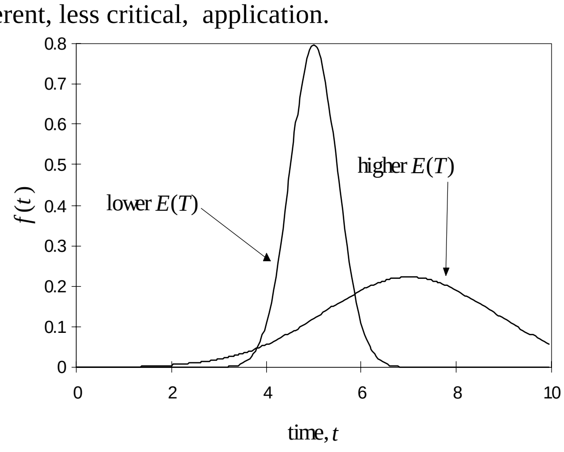

Figure 3: System Time-to-Failure Density Functions.

This interpretation of risk is also very applicable to other systems. Consider a consumer good (e.g., microwave oven, radio) where the customer will likely purchase a single unit and expects it to be maintenance-free for some extended (yet undefined) time period. A design which maximizes MTTF would be viewed less favorable, if some tangible subpopulation consistently fails early, compared to a design with a lower MTTF but highly likely to operate some extended period of time without failure. The design which maximizes that extended period of time would be the best.

These formulations are more realistic and offer benefits to risk-averse designers by explicitly considering risk. However, as they become more realistic, they also become more difficult, primarily because there are not closed-form expressions to calculate $R_{1-\alpha}(t_o,\mathbf{x})$ or $T_{1-\alpha}(\mathbf{x})$.

The redundancy allocation problem for series-parallel systems has proven to be difficult. Chern (Ref. 1)showed that the problem is NP-hard. Many different optimization approaches have been used to determine optimal or "very good" solutions. Previous research involving this problem can be classified based on the solution approach as dynamic programming, integer programming, mixed integer and nonlinear programming, or heuristics. An overview and summary of work in this area is presented in Tillman, Hwang and Kuo (Ref. 2)

Fyffe, Hines and Lee (Ref. 3), and Nakagawa and Miyazaki (Ref. 4) used dynamic programming to solve the problem. Ghare and Tayler (Ref. 5), and Bulfin and Lee (Ref. 6) used integer programming. For each of these formulations, the problem was to optimize system reliability. Gnedenko and Ushakov (Ref. 7) developed a heuristic to determine design configurations to maximize the median system time-to-failure. Risk was not considered in any of these formulations. Painton and Campbell (Ref. 8) solved a closely related problem using a genetic algorithm (GA) and Monte Carlo simulation which did consider risk.

3. *ALGORITHM DEVELOPMENT*

Separate algorithms were developed when the problem involved $R_{1-\alpha}(t_o,\mathbf{x})$ or $T_{1-\alpha}(\mathbf{x})$ as a performance measure. In either case, it could be the objective function or included as a constraint.

The algorithm to maximize $R_{1-\alpha}(t_o,\mathbf{x})$ assumed that component reliability was not known precisely for the available component types. This happens, for example, if it is known that the component time-to-failure follows a Weibull distribution but the shape and scale parameters are not known explicitly and are described by some prior distribution. The component independence assumption, applied probability theory and the central limit theory were used to show that the lognormal distribution accurately describes system reliability, to estimate distribution parameters and then to estimate $R_{1-\alpha}(t_o,\mathbf{x})$. Possible solution vectors, x, were searched using a GA. This algorithm is described in Ref. 9.

A similar algorithm was developed to maximize $T_{1-\alpha}(\mathbf{x})$. System and component time-to-failure are defined as follows,

T = system time-to-failure

$$= \min_i \left(\max_j T_{ij} \right) \quad \text{if } k_i = 1$$

T_{ij} = time-to-failure for the j^{th} component used in subsystem i

$T_{ij} \sim$ Weibull$(\lambda_{ij},\beta_{ij})$, $F(t) = 1 - \exp(-\lambda_{ij}(t)^{\beta_{ij}})$

It is difficult to solve this problem directly because, for any non-trivial problem, there is no closed-form expression to compute $T_{1-\alpha}(\mathbf{x})$. Monte Carlo simulation could be used as a function evaluator within a GA to compute this lower-bound, as was done by Painton and Campbell (Ref. 8). However, this can be inefficient considering that a separate function evaluation is required for every prospective solution considered and a GA must evaluate many different solutions each generation for an extended number of generations.

Instead, the problem was transformed to the following equivalent formulation by adding reliability to the constraint set.

Problem P1′: max t'

subject to $R(t',\mathbf{x}) \geq 1-\alpha$

$$\sum_{i=1}^{s} C_i(\mathbf{x}_i) \leq C$$

$$\sum_{i=1}^{s} W_i(\mathbf{x}_i) \leq W$$

$x_{ij} \in (0, 1, 2, ...)$

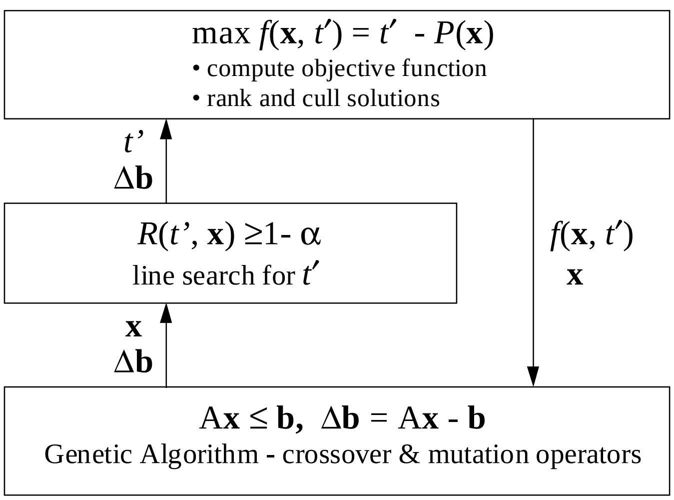

Figure 4: Algorithm For Problem P1′.

The t' which maximizes Problem P1′ is equal to the $T_{1-\alpha}(\mathbf{x})$ which then maximizes the original problem. The algorithm to solve this equivalent problem is depicted in Figure 4. Starting at the bottom of the figure, the GA crossover and mutation operators are used to search over possible solution vectors, **x**. The objective function value, t', is then found for each prospective solution via a Newton-Raphson search. The solutions are then ranked and culled by the t' values (plus a penalty for violated constraints). The process then repeats itself for some pre-selected number of generations or until the GA converges.

This algorithm avoids simulation and is therefore more efficient than other solution approaches which have been applied to problems of this type. The GA only operates on the **x** decision variables and finds t' in an efficient search which makes the reliability constraint tight.

For both algorithms, a GA was selected to search over possible solution vectors, **x**, and find the best solution. GAs are stochastic search algorithms patterned after natural selection in biological evolution as described by Holland (Ref. 10). The GA approach is very flexible, can accommodate both discrete and continuous functions, and can investigate large search spaces. Deficiencies with this approach are that there are several search parameters which must be found experimentally, and the GA can not guarantee convergence to the optimal solution, although it has consistently demonstrated that it provides good solutions to reliability problems. A GA specifically developed for reliability design problems was described by Coit and Smith (Ref. 11).

4. *EXAMPLE PROBLEMS*

Optimization of system designs considering risk profiles are demonstrated on two small illustrative example problems. The first problem is to minimize system cost when there is a reliability constraint but uncertainty with the component reliability estimates. The second example is to maximize $T_{1-\alpha}(\mathbf{x})$ at two different constraint levels with Weibull distributions known for the components’ time-to-failure.

The component parameters for the first example are presented in Table 1 (tables are at the end of the text). The problem is for a system with three subsystems and eight component choices per subsystem. The table shows the component reliability (r_{ij}), reliability variance (σ_{ij}^2) and three physical parameters labeled a_1, a_2 and a_3. The system-level values, A_i, are simply a sum of the component a_i. The objective is to identify the minimum cost configuration with a system-level reliability constraint of 0.97 and constraints on A_1, A_2 and A_3 of 40, 50 and 50. Physical interpretations of A_1, A_2 and A_3 were not provided but they could be upper limits on weight, size or some other physical or performance parameter. The system was optimized for risk levels of .50, .10 and .01.

Ten GA trials were performed for each risk level. The best solutions found are presented in Figure 5. These results are very interesting because the recommended solutions differ dramatically. As the designs situations become riskier, the design strategies change. This is particularly important because most optimization algorithms assume that designers are risk-neutral and will produce designs similar to the α=.50 design. Yet most real designers are risk-averse and desire systems closer to the α=.10 or .01 design.

The algorithm was next applied on two sample problems for both a risk-neutral (α = .50) and risk-averse (α = .05) designer. The problem is for a system with 14 subsystems. The component choices and their Weibull parameters are given in Table 2. Corresponding component weight and cost are given in Ref. 3, where this problem was first published. The size of the search space for this problem is $>10^{33}$. The two problems are to maximize $T_{1-\alpha}(\mathbf{x})$ with a system-level weight constraint of 191 and 159 respectively. There is a cost constraint of 130 for both problems.

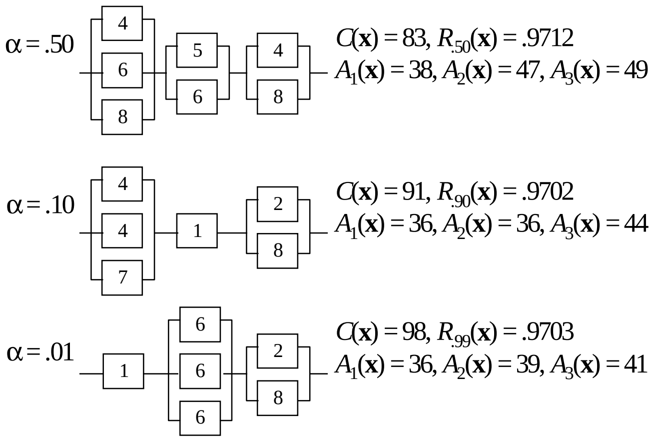

Figure 5: Problem Solutions For Example 1.

The table reveals some interesting comparisons. The component MTTF values vary by a factor of 100 for some of the subsystems. This would seem to imply that the components with the lower MTTF are less advisable choices, and in general, this is true. However, if the objective is to maximize the reliability for a relatively small mission time or the lower-bound on system time-to-failure is much smaller than the individual component MTTF values, then the component with the lower MTTF may actually be a better choice. This will be particularly true if the component choices with the higher MTTF are prone to infant mortality failures. They can very likely be poor choices for risk-averse design situations associated with relatively short system life.

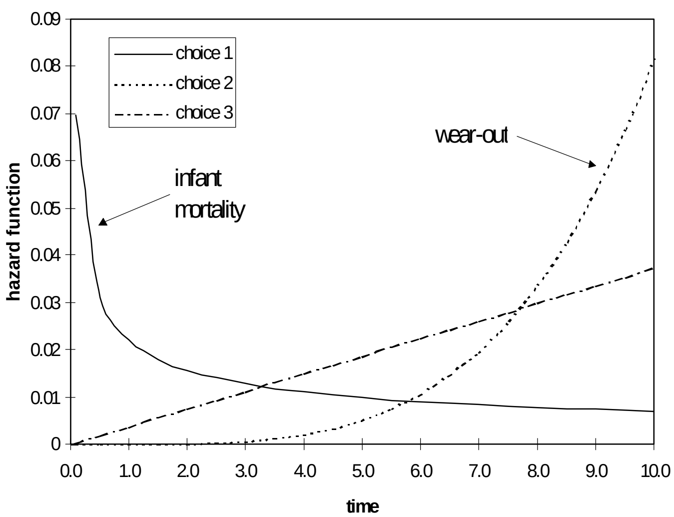

Figure 6: Subsystem 4 - Component Hazard Functions.

Figure 6 shows the hazard function for the three component choices available for subsystem 4. Notice that for short mission times, the rate of failure for the first choice is much higher than the other choices, even though this is the alternative with the highest mean-time-to-failure. Alternatively, the second choice rapidly wears out and generally fails early, but it fails less often during the very earliest time periods.

Ten GA runs were made for each different problem and α-level. Each GA run did not converge to the same solution, but even in the worst case, the GA solution had an objective function value very close to the best found. The most important observation is that the recommended design configurations are very different depending on the risk-level. This observation is demonstrated by the results shown in Figures 7 and 8. Figure 7 pertains to the least constrained example (W_o=191), and shows the lower-bound on system time-to-failure for three different design configurations as a function of α. The three different designs are the optimal solution which maximizes reliability (at time equal to 10) and the GA solutions for α values of 0.50 and 0.05 respectively. Figure 8 pertains to the most constrained example (W_o=159), and presents analogous information.

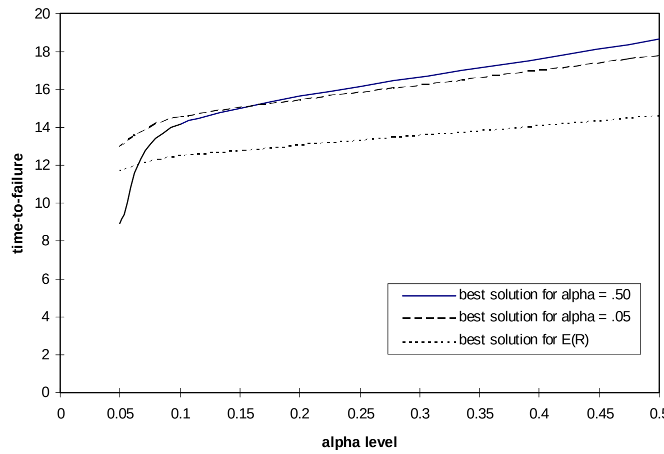

Figure 7: Comparison of Best Solution (W_o=191).

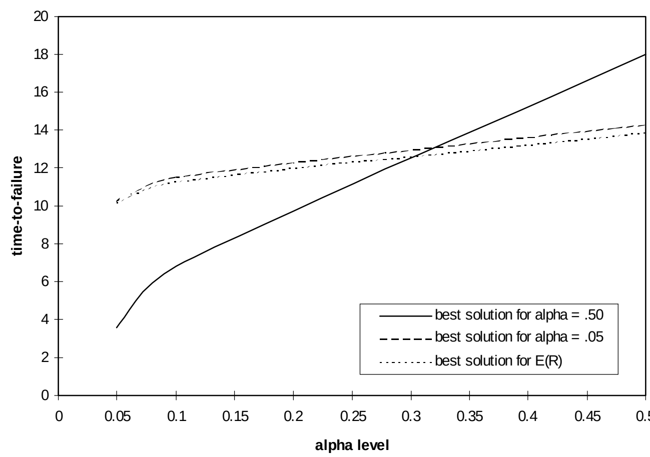

Figure 8: Comparison of Best Solutions (W_o=159).

For the example with W_o=191, the solution which maximizes reliability is not effective at maximizing a lower-bound on time-to-failure at any α-level. For the example with W_o=159, the solution which maximizes reliability is effective for lower α-levels, but not at higher α-levels, and is dominated by competing designs at all levels.

Also interesting is the that the solution which maximizes a lower-bound on time-to-failure for α=.50 is a poor choice for maximizing the lower-bound for α=.05, and similarly, the reverse argument is also true. Therefore, the amount of risk associated with a particular problem should affect design strategies. The optimal design configuration should be different if designers are risk-neutral and simply want to maximize the median system life, or alternatively, if the designers are risk-averse and want to assure that a great majority of the products will survive some minimal time period.

5. *DISCUSSION*

In certain factories using complex heavy machinery, component failures are relatively common, personnel are ready to respond and they can be repaired efficiently. In this case, the user desires to maximize the expected or median life of the system, and is generally risk-neutral. For a risk-neutral designer, an α of 0.50 is appropriate. Alternatively, if component failure is relatively rare, but the implications of failure are very dire, then the user will be risk-averse. The degree at which the particular designer or designer is risk-averse is controlled by a quantitative risk level (α). For a risk-averse designer, a low value of α should be used where α represents the probability that a randomly selected system will have a reliability lower than the constraint value or objective function.

The problem formulations and solution algorithms which were demonstrated allow for the explicit consideration of risk. Often, estimates of component reliability are not known precisely. It is not possible to measure reliability in a manner similar to a weight or a resistance. It is necessary to conduct life tests or to observe the performance of fielded systems using the same part. For highly reliable components, there often must be a large sample size and a lengthy test duration to determine an accurate estimate of reliability. Since this is not always possible, and even when it is possible there are still sources of variability, the component reliability should be considered as a random variable. Therefore, system reliability is also a random variable and it becomes necessary to determine a lower-bound on system reliability.

When a lower-bound on reliability or time-to-failure is maximized, the user is required to specify an appropriate risk level (α). There are typical values which are often used. For example, a risk neutral designer would use an α of .50, a moderately risk-averse designer may likely use an α of .10 or .05, and a highly risk-averse designer may use a value of .01 or .001. However, these are simply general guidelines. It would be very interesting to apply utility theory and decision analyses to many different engineering design problems and assess the risk levels which maximizes the designer's overall utility while considering the implications of an unreliable system, the benefits of a highly reliable system, etc.

ACKNOWLEDGMENT

The second author gratefully acknowledges financial support from the National Science Foundation CAREER grant DMI-9502134.

REFERENCES

1. M. S. Chern, "On the Computational Complexity of Reliability Redundancy Allocation in a Series System", *Operations Research Letters*, Vol. 11, (June, 1992), pp. 309-315.
2. F. A. Tillman, C. L. Hwang and W. Kuo, *Optimization of System Reliability*, (New York: Marcel Dekker, 1980).
3. D. Fyffe, W. Hines and N. Lee, "System Reliability Allocation and a Computational Algorithm", *IEEE Transactions on Reliability*, Vol. R-17, (June, 1968), pp. 64-69.
4. Y. Nakagawa and S. Miyazaki, "Surrogate Constraints Algorithm for Reliability Optimization Problems With Two Constraints", *IEEE Transactions on Reliability*, Vol. R-30, (June, 1981), pp. 175-180.
5. P. M. Ghare and R. E. Taylor, "Optimal Redundancy for Reliability in Series System", *Operations Research*, Vol. 17, (September, 1969), pp. 838-847.
6. R. L. Bulfin and C. Y. Liu, "Optimal Allocation of Redundant Components for Large Systems", *IEEE Transactions on Reliability*, Vol. R-34, (August, 1985), pp. 241-247.
7. B. Gnedenko and I. Ushakov, *Probabilistic Reliability Engineering*, (New York: John Wiley & Sons, 1995), pp. 505-511.
8. L. Painton and J. Campbell, "Genetic Algorithms in Optimization of System Reliability", *IEEE Transactions on Reliability*, Vol. R-44, (1995), pp. 172-178.
9. D. W. Coit and A. E. Smith, "Stochastic Formulations of the Redundancy Allocation Problem," *Proceedings of the Fifth Industrial Engineering Research Conference*, Minneapolis, MN, May 1996, pp. 459-463.
10. J. Holland, *Adaptation in Natural and Artificial Systems*, (U. of Michigan Press, 1975).
11. D. W. Coit and A. E. Smith, "Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm", *IEEE Transactions on Reliability*, Vol. 45, (June, 1996), pp. 254-260.

BIOGRAPHIES

David W. Coit, Ph.D.
Rutgers University
Dept. of Industrial Engineering
PO Box 909
Piscataway, NJ 08855-0909 USA

David Coit is an Assistant Professor of Industrial Engineering at Rutgers. He has degrees from Cornell, RPI and University of Pittsburgh and previously worked for 12 years in reliability engineering at IIT Research Institute. His interests are in reliability optimization, probabilistic modeling of engineering systems and genetic algorithms. He is a member of IEEE, IIE and INFORMS.

Alice E. Smith, Ph.D., P.E.
Department of Industrial Engineering
University of Pittsburgh
1031 Benedum Hall
Pittsburgh, PA 15261 USA
Internet (email): aesmith@engrng.pitt.edu

Alice Smith is Associate Professor of Industrial Engineering. After ten 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 and optimization of complex systems using computational intelligence techniques, and her research has been sponsored by Lockheed Martin Corporation, ABB Daimler-Benz Transportation, the Ben Franklin Technology Center of Western Pennsylvania, the National Institute of Standards and the National Science Foundation, from which she was awarded a CAREER grant in 1995. Her work in design optimization when considering system reliability has appeared in *IEEE Transactions on Reliability*, *INFORMS Journal on Computing*, *Computers & Operations Research* and *Computers & Industrial Engineering*. Dr. Smith is a senior member of IEEE, IIE and SWE, and a member of ASEE and INFORMS.

Table 1: Example 1 Problem Parameters.

	subsystem 1						subsystem 2						subsystem 3					
j	r_{ij}	σ_{ij}^2	c_{ij}	a_1	a_2	a_3	r_{ij}	σ_{ij}^2	c_{ij}	a_1	a_2	a_3	r_{ij}	σ_{ij}^2	c_{ij}	a_1	a_2	a_3
1	.99	.005	30	7	9	5	.99	.005	29	8	7	4	.99	.005	24	8	5	9
2	.90	.050	14	9	10	7	.90	.050	22	8	8	10	.90	.050	22	6	4	6
3	.86	.070	13	8	9	8	.86	.070	19	10	9	9	.86	.070	20	10	4	7
4	.84	.080	10	5	7	9	.84	.080	18	6	5	8	.84	.080	16	6	9	4
5	.83	.085	11	4	9	7	.83	.085	17	5	10	9	.83	.085	15	5	10	10
6	.82	.090	10	4	6	5	.82	.090	11	5	7	6	.82	.090	12	9	8	4
7	.79	.105	9	6	7	7	.79	.105	10	8	8	7	.79	.105	12	7	9	6
8	.75	.125	8	7	4	7	.75	.125	9	10	8	6	.75	.125	11	6	4	9

Table 2: Input Parameters for Example 2.

	choice 1			choice 2			choice 3			choice 4		
i	λ_{ij}	β_{ij}	MTTF	λ_{ij}	β_{ij}	MTTF	λ_{ij}	β_{ij}	$E(T)$	λ_{ij}	β_{ij}	MTTF
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