



A Computational Complexity Analysis of an Algorithm for Reliability Optimization

Benedict K. Nmah^{1*}

¹Department of Mathematics, Morehouse College, Atlanta, GA 30314, USA.

Article Information

DOI: 10.9734/BJMCS/2015/14835

Editor(s):

(1) Sergio Serrano, Department of Applied Mathematics, University of Zaragoza, Spain.

Reviewers:

(1) Thomas Chambrion, Institut Elie Cartan, Universit de Lorraine, France.

(2) Anonymous, China.

(3) Anonymous, South Africa.

(4) Anonymous, Brazil.

Complete Peer review History:

<http://www.sciencedomain.org/review-history.php?iid=932&id=6&aid=7902>

Original Research Article

Received: 23 October 2014

Accepted: 06 January 2015

Published: 29 January 2015

Abstract

Least-cost allocation of redundancy within subsystems of a series-parallel system to meet a requirement for system reliability may be modelled as a nonlinear integer program in which the number of variables is equal to the number of subsystems. The data are the system requirement, the reliabilities of the identical components within each subsystem and the costs of these components. The model has a linear cost function as its objective and a single non-linear constraint that enforces the requirement for system reliability. The continuous counterpart of the discrete problem can be solved by solving a polynomial equation. The main result of this paper is a bound on the effort required to obtain from the solution of the continuous relaxation the optimal numbers of redundant components for the subsystems.

Keywords: Discrete optimization, nonlinear integer programming, reliability optimization, computational complexity, redundancy allocation, optimization model.

2010 Mathematics Subject Classification: 90C60

1 Introduction

The literature on system reliability study is satiated with various definitions of system reliability. The common theme in all the definitions is that the reliability of a system is essentially the probability

*Corresponding author: E-mail: benedict.nmah@morehouse.edu

that the system will successfully perform its intended function within a specified interval of time. The reliability of a system is an inherent attribute, just as the system's capacity or power rating.

The design process of any system has two major objectives. The first objective is that the system must be designed to meet its functional goals, and the second objective is that the system must be designed to successfully perform its intended function in a specified, and often severe, environment [1]. Undoubtedly, the immeasurable inconveniences that any failure of a necessary system would cause a society underscore the importance of reliability study. The latter of the two major goals of the design process requires designing reliability into the system. Usually the failure characteristics of some systems are used to determine the economic feasibility of the system.

Reliability allocation is the process by which designers translate requirements for system reliability into reliability goals for subsystems and components [2]. Such goals may be constrained by limitations on space, weight, power consumption or cost. Early in the design process, a complex system may be conceived as a series of m independent subsystems, with reliability goals R_1, R_2, \dots, R_m . To achieve a required value, R_r , for the reliability of the entire system, the subsystem goals are linked by the constraint $\prod_{i=1}^m R_i \geq R_r$ [3]. To realize a subsystem goal, designers consider various arrangements

of components. When resilience in response to random failures is an important consideration, as in many electronic subsystems, redundancy among components is a frequently used design strategy. Specifically, active redundancy of components in the i^{th} subsystem is achieved with x_i components that operate simultaneously, fail at the same rate and are not repaired [4]. The subsystem works as long as at least one of the components works. Given r_i , the common reliability value of the components, the designer considers values, x_i , for the number of components so that $1 - (1 - r_i)^{x_i} \geq R_i$. However, redundancy enhances reliability only with additional demands for capital, power, space, weight and maintenance [5]. As a result, designers evaluate trade-offs among redundancy and other constrained resources. Some of these trade-offs maybe modelled as an integer program of the form

$$(1.1) \quad \begin{cases} \text{Min } \sum_{i=1}^m c_i x_i \\ \text{Subject to } \prod_{i=1}^m (1 - \rho_i^{x_i}) \geq R_r \\ x_i \geq 1 \text{ for } i = 1, 2, 3, \dots, m \\ x_i \text{ integral for } i = 1, 2, 3, \dots, m. \end{cases}$$

Here, the values r_i ($i = 1, 2, \dots, m$) are replaced by their complements $\rho_i = 1 - r_i$, component unreliability for the i^{th} stage; R_r is the minimum required system reliability, $0 < R_r < 1$. Each of the objective coefficients c_i is assumed to be strictly positive. These coefficients model a trade-off between reliability and another scarce resource in the design process.

Feasible solutions of this nonlinear integer programming problem are readily available. For instance, one can allocate the system reliability goal equally among the subsystems and then set

$$x_i = \left\lceil \frac{\ln(1 - \sqrt[m]{R_r})}{\ln(\rho_i)} \right\rceil, \text{ for } 1 \leq i \leq m.$$

Since all objective coefficients are positive, candidates for an optimal integral solution are limited to the finite set of feasible solutions with objective values that do not exceed the cost of this feasible solution. As a result, the nonlinear integer programming problem has at least one optimal solution.

The integrality constraints of (1.1) can be relaxed by requiring only $x \in R_+^m$. Nmah [6] showed that this "continuous" nonlinear programming problem has a unique optimal solution. Denote by x^*

the optimal, probably nonintegral, solution of the “continuous” nonlinear programming problem. We can obtain a lower bound, β_0 , on the objective value of the integer programming problem by defining

$$\beta_0 = \left\lceil \sum_{i=1}^m c_i x_i^* \right\rceil.$$

We can also obtain a feasible solution of the integer programming problem by taking the upper ceiling of each coordinate of x^* to get an integral vector $x^{(*I)}$. If x is any optimal integral solution of (1.1), then x must be in an intersection

$$K_m(\beta) := \{x \in \mathbb{N}^m \mid \sum_{i=1}^m c_i x_i = \beta, x \geq 1\},$$

of a hyperplane and the positive cone, where β is an integer between

$$\sum_{i=1}^m c_i x_i^* \quad \text{and} \quad \sum_{i=1}^m c_i x_i^{(*I)}.$$

The purpose of this paper is to bound the worst-case effort to locate all optimal solutions of the integer program (1.1) on the hyperplanes $K_m(\beta)$.

2 Related Work

Optimization models provide two complementary perspectives on the trade-off between requirements for system reliability and demands on scarce resources. These perspectives have sustained a long history, documented in [7], [8] and [9], of work on various criteria for reliability, on models of complex systems, and on several implementations of redundancy. Solutions of these models have also demonstrated many exact and heuristic methods for discrete and nonlinear optimization. From one perspective, the trade-off is addressed by determining the highest level of reliability to be achieved with fixed levels of resources. In this setting, the optimization model has a reliability criterion as the objective, while the resources are modeled as one or more constraints. From another perspective, a reliability requirement is fixed and the optimization model has a cost criterion as the objective. The model proposed in (1.1) is formulated from this perspective, with a specific focus on redundancy as the design strategy by which reliability within subsystems is achieved.

Majety, Dawande and Rajgopal [10] consider a variant of (1.1) that models component cost-reliability relationships that do not fit a particular functional form. Yalaoui, Châtelet and Chu [11] model series-parallel systems in which a single subsystem may be designed with components of different types and reliabilities. Although the reliability constraint in (1.1) can be transformed by logarithms into a constraint in which the original decision variables appear in separate terms, it cannot be readily re-written as a constraint linear in the original decision variables [10]. Djerdjour and Rekab [12] rewrite a separable version of the constraint as a piece-wise linear function of the original variables. With upper and lower bounds on the original variables, they reformulate the model as a linear program with a larger number of binary variables and then solve the model with a branch-and-bound algorithm. With upper and lower bounds on each decision variable, Sun, Ruan and Li [13] use the primal-dual relationships of nonlinear programming to study both types (system reliability as the objective and system reliability as a constraint) of redundancy allocation problems.

When the integrality constraints of (1.1) are relaxed to allow decision variables that are real-valued and positive, Nmah [14] showed that the continuous relaxation has a unique optimal solution, x^* , for which the reliability constraint is satisfied as an equation and gave an algorithm based on a Lagrange multiplier to produce this solution. The objective value of x^* provides a lower bound for

the optimal value of the integer program (1.1). An upper bound is the objective value of the feasible solution obtained rounding up each of the values of the decision variables of the optimal solution of the continuous problem. A search of a finite number of finite sets of the form $K_m(\beta)$ for β between the upper and lower bounds produces all of the optimal solution of the integer program.

Liu [15] considers a model in which both the levels of redundancy and the component reliabilities are decision variables. The constraint on system reliability is augmented by resource constraints for the individual subsystems, as well as by upper and lower bounds on the reliability of the individual subsystems. Falk, Singpurwalla and Vladimirsky [16] formulate and solve, with the use of Lagrange multipliers, a reliability allocation model in which the decision variables are continuous variables, corresponding to the reliabilities of the subsystems. Their model proposes goals, rather than designs, for systems in which the configuration of subsystems may be more complex than the series of independent subsystems considered here. Ebeling [17] proposed a reliability allocation model in which continuous decision variables represent improvements over the reliabilities of independent components or subsystems in an initial design. When the objective is a quadratic function of these improvements, some cases of the model can be solved by an iterative procedure that updates a Lagrange multiplier.

3 Description of the Algorithm

Let the nonlinear integer programming problem be as in (1.1) and denote $\prod_{i=1}^m (1 - \rho_i^{x_i})$ by $R_s(x)$. For the relaxed problem, where $x \in R_+^m$, consider the Lagrangian:

$$L(x, \lambda) = \sum_{i=1}^m c_i x_i + \lambda g(x)$$

with $g(x) = R_r - R_s(x)$.

Nmah [14] showed that the equations

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

can be solved explicitly to produce the optimal solution. In fact, the equation $\nabla_\lambda L(x^*, \lambda^*) = 0$ can be written as

$$\prod_{i=1}^m (-\lambda \ln(\rho_i) R_r) = R_r \prod_{i=1}^m (c_i - \lambda \ln(\rho_i) R_r).$$

This equation is a polynomial of degree m in λ . Its unique positive root determines the optimal value of the Lagrange multiplier. Then the equations

$$\nabla_x L(x^*, \lambda^*) = 0$$

determine the optimal values of the decision variables as

$$x_i^* = \frac{\ln\left(\frac{c_i}{c_i - \lambda^* \ln(\rho_i) R_r}\right)}{\ln(\rho_i)}.$$

Thus the method for solving the nonlinear integer program consists of searching hyperplanes in order of increasing values of β , beginning with any lower bound (such as β_0) for the optimal objective value of the integer program. Two things are worth noting here. First, a hyperplane is searched to

find integral vectors, and secondly, once such an integral vector is found, it is tested to determine whether or not it satisfies the reliability requirement. If it does, it is an optimal solution of the integer program, and the algorithm stops. If one continues to search the rest of this hyperplane, one can find all of the optimal solutions of a problem with multiple solutions. No additional hyperplane needs to be searched. On the other hand, if it does not satisfy the reliability requirement, one looks for another integral vector on the hyperplane and tests it to determine whether or not it satisfies the reliability requirement. Given this observation, there is a need to bound the number of hyperplanes that must be searched.

Proposition 3.1. *The number of hyperplanes that may be searched to find optimal integral solutions is bounded.*

Proof. Suppose that β^* is the optimal objective value of the “continuous” nonlinear programming problem. Then the value of $x^{(*I)}$ is no greater than $\beta^* + \sum_{i=1}^m c_i$. Since $\beta_0 \geq \beta^*$, the number of

hyperplanes to be searched is no greater than $\sum_{i=1}^m c_i$. □

4 Worst-Case Computational Complexity of the Algorithm

We now show that the worst-case computational complexity of the algorithm is bounded. For a hyperplane defined by $\sum_{i=1}^m c_i x_i = \beta$, a worst-case search is one that produces no integral solution of the reliability constraint. Nmah [6] explained how to produce integral vectors on the hyperplane. However, our goal in this paper is only to investigate the worse-case computational complexity of the algorithm, and not its implementation. Specifically, the phrase, worst-case computational complexity, is used to describe the largest number of feasible solutions of a problem of the form

$$(4.1) \quad \begin{cases} \sum_{i=1}^m c_i x_i = \beta \\ \text{s.t. } x \geq 1 \text{ and } \beta \in \mathbb{N} \end{cases}$$

that must be listed when no integral feasible solution of $R_s(x) \geq R_r$ is found.

Furthermore, the largest number of solutions for this type of problem occurs when all the cost coefficients are identically equal to 1. This observation is justified by noting that the equation

$\sum_{i=1}^m c_i x_i = \beta$ where the c_i , ($i = 1, 2, 3, \dots, m$) and β are positive integers has solutions over the integers iff $gcd(c) | \beta$. Therefore, we can assume without loss of generality that $gcd(c) = 1$ since factoring out the $gcd(c)$ gives another problem with identical solutions [6]. Moreover, if \tilde{x} solves $\sum_{i=1}^m c_i x_i = \beta$, then \tilde{y} , given by $\tilde{y}_i = c_i \tilde{x}_i$ ($i = 1, 2, 3, \dots, m$) also solves $\sum_{i=1}^m y_i = \beta$. Therefore, all

solutions of $\sum_{i=1}^m c_i x_i = \beta$ are included in those of $\sum_{i=1}^m x_i = \beta$. Thus, setting $c_i = 1$ for every i , ($i = 1, 2, 3, \dots, m$), (4.1) reduces to

$$(4.2) \quad \begin{cases} 1^T x = \beta, \text{ where } 1 \in \mathbb{R}^m \\ \text{s.t. } x \geq 1 \text{ and } \beta \in \mathbb{N} \end{cases}$$

and the number of solutions of (4.2) corresponds to the number of ways of choosing β items from a set with m elements, so that x_i items of type i are chosen, ($i = 1, 2, 3, \dots, m$). This number is equal to the number of β combinations with repetition allowed from a set with m elements, and is given by

$$C_{\beta-m}^{\beta-1} \text{ where } C_{\beta-m}^{\beta-1} = \binom{\beta-1}{\beta-m}.$$

Proposition 4.1. *The worst-case computational complexity of the algorithm is $\left(\sum_{i=1}^m c_i\right) \times \frac{m}{\beta} \times \binom{\beta}{m}$.*

Proof. Noting the foregoing, the proof has been established. The number of hyperplanes to be searched is at most $\sum_{i=1}^m c_i$; on a single hyperplane, the number of vectors to be tested is at most $C(\beta-1, m-1)$. On the hyperplanes to be searched, the integer will not exceed the best upper bound for (1.1) available at the start of the search. Since $(\beta! = \beta \times (\beta-1)!) \implies ((\beta-1)! = \frac{\beta!}{\beta})$, and similarly for $(m-1)!$, we have,

$$\left(\sum_{i=1}^m c_i\right) \times \binom{\beta-1}{\beta-m} = \left(\sum_{i=1}^m c_i\right) \times \frac{m}{\beta} \times \binom{\beta}{m}.$$

Thus, we have shown that the worst-case computational complexity of the algorithm is

$$O\left(\left(\sum_{i=1}^m c_i\right) \times \frac{m}{\beta} \times \binom{\beta}{m}\right).$$

□

5 Conclusions

The redundancy allocation model (1.1) can be solved in two phases. First, the constraints can be relaxed so that the decision variables are real and strictly positive. For strictly positive objective coefficients, the reliability constraint is satisfied as an equation at any optimal solution. The only candidates for optimal solutions of the continuous problems are solutions of the Lagrangian equation $L(x, \lambda) = 0$. In fact, for data $c_i, \rho_i : 1 \leq i \leq m$ and R_r , this equation has a unique solution (x^*, λ^*) in which the values of the decision variables are determined by the data and the optimal multiplier λ^* . The value of the optimal multiplier is the value of the unique positive root of an m^{th} degree polynomial, and the root can be obtained with Newtons method. At the end of this phase, a lower bound, β_0 , for the optimal cost of the integer programming model is obtained as the integral ceiling of the optimal value of the continuous problem. For the application to redundancy allocation, completion of this phase may provide useful information about the feasibility of the project being modelled. For example, early in a design project, the model can be used to assess the cost to complete the project. If the bound β_0 is deemed too high, then the project, the reliability requirement or the expectations for resources can immediately be re-evaluated [4].

In the second phase, an upper bound on the optimal objective value, which differs from β_0 by at most $\sum_{i=1}^m c_i$, can be computed by rounding up each decision variable. Integral vectors in finite sets of the form $K_m(\beta)$, beginning with β_0 and ending with the first successful outcome, are tested for conformance with the reliability requirement. A complete search of the set that yields the first

success will produce all optimal integral solutions when the model has multiple optimal solutions. In this paper, we have bounded the worst-effort needed to complete this phase.

Acknowledgment

The author acknowledges the referees for valuable comments and suggestions that helped to improve the manuscript.

Competing Interests

The author declares that no competing interests exist.

References

- [1] Birolini A. Reliability engineering: theory and practice. Berlin. Springer; 2010.
- [2] Kapur KC. Reliability and maintainability. In: Salvendy G, editor. Handbook of Industrial Engineering: Technology and operations management. 3rd ed. New York: Wiley; 2001.
- [3] Pecht M. Reliability, maintainability and availability. In: Sage AP, Rouse WB, editors. Handbook of systems engineering and management. 2nd ed. Hoboken, NJ: John Wiley & Sons; 2009.
- [4] Hecht H. System reliability and failure prevention. Boston: Artech House; 2004.
- [5] Smith DJ. Reliability, maintainability and risk. 8th ed. Amsterdam: Elsevier Butterworth-Heinemann; 2011.
- [6] Nmah B. Optimizing system reliability with integer programming. New Mexico State University: UMI Dissertations Services, Bell & Howell Company; 1997.
- [7] Tillman FA, Hwang C, Kuo W. Optimization techniques for system reliability with redundancy: a review. *IEEE Trans. Reliability*. 1977;26(3):148-155.
- [8] Kuo W, Prasad R. System reliability optimization: an overview. In: Soyer R, Mazzuchi TA, Singpurwalla ND, editors. Mathematical reliability: an expository perspective. Boston: Kluwer Academic Publishers; 2004.
- [9] Kuo W, Wan R. Recent advances in optimal reliability allocation. *IEEE Trans. Systems, Man and Cybernetics A*. 2007;37(2):143-156.
- [10] Majety SRV, Dawande M, Rajgopal J. Optimal reliability allocation with discrete cost-reliability data for components. *Operations Research*. 1999;47(6):899-906.
- [11] Yalaoui A, Chtelet E, Chu C. A new dynamic programming method for reliability & redundancy allocation in parallel-series systems. *IEEE Trans. Reliability*. 2005;54(2):254-261.
- [12] Djerdjour M, Rekab K. A branch and bound algorithm for designing reliable systems at a minimum cost. *Applied Math. & Computation*. 2001;118(2-3):247-259.
- [13] Sun X, Ruan N, Li D. An efficient algorithm for nonlinear integer programming problems arising in series-parallel reliability systems. *Optimization Methods and Software*. 2006;21(4):617-633.
- [14] Nmah B. Optimal allocation of redundancy in a mixed system. *International Journal of Pure and Applied Mathematics*. 2011;73(3):255-265.
- [15] Liu GS. A combination method for reliability-redundancy optimization. *Engineering Optimization*. 2006;38(4):485-499.

- [16] Falk JE, Singpurwalla ND, Vladimirsky YY. Reliability allocation for networks and systems. SIAM Review. 2006;48(1):43-65.
- [17] Ebeling C. An introduction to reliability and maintainability engineering. 2nd ed. Long Grove, IL: Waveland Press; 2010.

©2015 Benedict K. Nmah; This is an Open Access article distributed under the terms of the Creative Commons Attribution License <http://creativecommons.org/licenses/by/4.0>, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)

www.sciencedomain.org/review-history.php?iid=932&id=6&aid=7902