Interval Arithmetic Technique for Constrained Reliability Optimization Problems *

Humberto Muñoz and Ernst Pierre
Southern University and A&M College at Baton Rouge, LA

Abstract. This work presents an interval arithmetic technique to solve constrained reliability optimization problems arising in systems with unit redundancy. The technique finds efficiently the global minimum of the cost function under reliability constraints, and it also solves the corresponding dual problem. The results obtained with this technique are compared with those obtained with the classical Lagrange multiplier method and the branch-and-bound technique,which are very commonly used for both the redundancy allocation problem and the mixed integer-type reliability-redundancy allocation problem. Some illustrative examples are provided.

Keywords: Interval arithmetic, reliability optimization, integer programming, mixed integer programming, Lagrange multiplier method, Kuhn-Tucker conditions, Branch-and-bound, Interval Newton's method

1. Introduction

In most reliability optimization problems, the decision variables are the number of redundancies that are integers (integer programming or redundancy allocation problems), the component reliabilities that are real numbers (real programming or reliability allocation problems), or a combination of both (mixed integer programming or reliability-redundancy allocation problems). These problems have been studied in great detail for different reliability optimization techniques as the exact techniques: the Lagrange multiplier (LM) with Kuhn-Tucker conditions (Misra and Ljubojevic, 1973) and dynamical programming (Tillman et al., 1985), or the iterative methods: the branch-and-bound technique (Nakagawa et al. , 1978), and the heuristic method (Tillman et al., 1978). Only few techniques have demonstrated to be effective when applied to large scale nonlinear redundancy allocation problems , (Ramakumar, 1993), (Harunuzzaman and Aldemir, 1996), (Bulfin and Liu, 1985). Another drawback is that the solutions are non integers and hence the true optimal solution which must be integer is not guaranteed. An efficient method combining the Lagrange multiplier method and the branch-and-bound technique (LMBB) is proposed by (Way et al., 1987). In the other hand, interval techniques have proved to be effective solving nonlinear global optimization problems (Ratschek and Rokne, 1990), (Hansen, 1992), (Kearfott, 1996), (Muñoz, 2002). In this paper, an interval arithmetic technique will be introduced to solve reliability optimization problems. The results of this interval solution technique will be compared with those obtained from the LM and LMBB techniques.

* This work was possible with the support of OSP-SUBR

In this paper we are dealing with the solution of the following integer programming problems:

$$\begin{aligned} \text{Minimize } C(x)&, \\ \text{Subject to } \ R(x) &\geq R_0, \\ x &= (x_1, x_2, \ldots, x_n), \\ x_i &\geq 2, \\ x_i \text{ is integer for } i &= 1, \ldots, n, \end{aligned} \tag{1}$$

or

$$\begin{aligned} \text{Maximize } R(x)&, \\ \text{Subject to } \ C(x) &\leq C_0, \\ x &= (x_1, x_2, \ldots, x_n), \\ x_i &\geq 2, \\ x_i \text{ is integer for } i &= 1, \ldots, n \end{aligned} \tag{2}$$

where $C(x)$ and $R(x)$ are differentiable functions, and $0 \leq R_0 \leq 1$.

The content of this paper is as follows: In the next section we formulate the optimization problems arising in systems with unit redundancy. In Section 3, interval arithmetic notation is introduced. Interval optimization tools and an algorithm to solve (1) , (2) are presented in Section 4. Illustrative examples are shown in Section 5, and conclusions are given in Section 6.

2. Problem Formulation

There are different types of optimization problems related to systems with unit redundancy (Agrafiotis and Tsuokalas, 1994; Coit, 2001; Gurov et al., 1995).

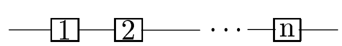

Figure 1. Original System of n units in series

An original system is shown in Figure 2, and it is assumed to consist of a number of n units, all of which must be working for the system to succeed. The total cost C_0 is obtained by $C_0 = \sum_{i=1}^{n} c_i$, where c_i is the cost of unit i, and the total system reliability R_0 is obtained by $R_0 = \prod_{i=1}^{n} p_i$, where p_i is the reliability of unit i.

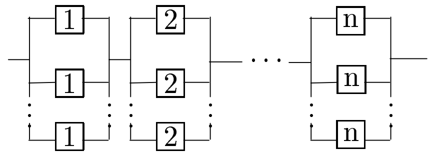

Figure 2. System with *Unit Redundancy Configuration*

If the individual units of the system are replicated, then we have *unit redundancy*, as shown in Figure 2.

Given the basic system with n different units as shown in Figure 2, we want to improve the overall system reliability to R, by using the unit redundancy with a minimum cost.

The two main questions to be addressed are:

a) What should be the minimum number of redundancies for the i-th unit, for $i = 1, 2, \ldots, n$, so that the system reliability is maximum?

b) What should be the number of redundancies for the i-th unit, for $i = 1, 2, \ldots, n$, so that the system cost is minimum?

Minimizing the System Cost

The constrained optimization problem is defined by

$$\begin{aligned} \text{minimize } C &= \sum_{i=1}^{n} c_i x_i, \\ \text{subject to } \; R &= \prod_{i=1}^{n} [1 - (1 - p_i)^{x_i}] \\ &\geq R_0, \end{aligned} \tag{3}$$

Maximizing the System Reliability

The constrained optimization problem is defined by

$$\begin{aligned} \text{maximize } R &= \prod_{i=1}^{n} [1 - (1 - p_i)^{x_i}] \\ \text{subject to } \; C &= \sum_{i=1}^{n} c_i x_i \leq C_0, \end{aligned} \tag{4}$$

where

C :	total system cost
C_0 :	the maximum required system cost
R :	system reliability
R_0 :	the minimum required system reliability
c_i :	cost of unit i.
x_i :	the number of units in parallel replacing the original unit i.
p_i :	reliability of unit i.

3. Interval Arithmetic Concepts

Real interval arithmetic was introduced in (Moore, 1962). Based on (Kearfott, 1996), *interval arithmetic operations* are defined by the following rules

$$[a,b]+[c,d]=[a+c,b+d]$$
$$[a,b]-[c,d]=[a-d,b-c]$$
$$[a,b]\cdot[c,d]=[\min\{ac,ad,bc,bd\},\max\{ac,ad,bc,bd\}]$$
$$[a,b]/[c,d]=[a,b]\cdot[1/d,1/c] \text{ if } 0\notin[c,d]$$

Let g be any predefined function in some programming language (like $\sin, \cos, \exp$, etc.). Then the *corresponding predefined interval function* IG is defined by the range of g over $\boldsymbol{y}$.

$$IG(\boldsymbol{y})=g(\boldsymbol{y})=\{g(y):y\in\boldsymbol{y}\}.$$

Interval arithmetic tools needed to solve (6) , (14) are established in Ratschek's work see (Ratschek and Rokne, 1990). The main interval arithmetic tool applied to optimization problems is the concept of an inclusion function. A function F is called an *inclusion function* for f if

$$f(\boldsymbol{y})\subseteq F(\boldsymbol{y}) \text{ for any } \boldsymbol{y}.$$

Let $f(x)$ be any expression in the variable $x\in\mathbb{R}^m$. Then the expression which arises if each occurrence of x in $f(x)$ is replaced by $\boldsymbol{x}$, if each predeclared function g in $f(x)$ is replaced by IG, and if the arithmetic operations in $f(x)$ are replaced by the corresponding interval arithmetic operations, is called the *natural interval extension* of $f(x)$ to $\boldsymbol{x}$ and it is denoted by $f(\boldsymbol{y})$. The following example illustrates interval arithmetic.

EXAMPLE 1. *Let* $f:\mathbb{R}^2\to\mathbb{R}$,*be the Rosenbrock function* $f(x,y):=100*(y-x^2)^2+(1-x)^2$. *Consider the interval vector* $\boldsymbol{X}:=(\boldsymbol{x},\boldsymbol{y})=([0.7,1],[0.0,0.3])$. *Then for* $x\in\boldsymbol{x}$ *and* $y\in\boldsymbol{y}$,

$$\begin{aligned}
f(x,y) &= 100*(y-x^2)^2+(1-x)^2\\
&\in 100*(\boldsymbol{y}-\boldsymbol{x}^2)^2+(1-\boldsymbol{x})^2\\
&= 100*([0.0,0.3]-[0.7,1]^2)^2+(1-[0.7,1])^2\\
&= 100*([0.0,0.3]-[0.49,1])^2+[0.0,0.3]^2\\
&= 100*([-1.0,-0.19]^2+[0.0,0.091]\\
&= [3.61,100.09].
\end{aligned}$$

4. Solution Methods

4.1. The Lagrange Multipliers Method, the Kuhn-Tucker Conditions and Newton's Method

This technique provides the exact solution of constrained optimization problems. Given the function $\phi : \boldsymbol{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}$ and constraints $C(x) = (c_1(x), \ldots, c_m(x))^T : \mathbb{R}^n \rightarrow \mathbb{R}^m$, for $x = (x_1, \cdots, x_n) \in \boldsymbol{X}$ solve

$$\begin{aligned} &\text{Minimize } \phi(x) \\ &\text{subject to } \; c_i(x) \; \leq \; b_i, \quad i = 1, \cdots, m. \end{aligned} \tag{5}$$

The functions $\phi(x)$ and $c_i(x)$ can be nonlinear functions. The Lagrange multipliers technique transforms the constrained optimization problem (5) in an unconstrained optimization problem by introducing the Lagrange multipliers. The new objective function, called the Lagrangian, becomes

$$\begin{aligned} \text{Minimize } L(x, \lambda) \; &= \; \phi(x) - \sum_{i=1}^{m} \lambda_i (c_i(x) - b_i) \\ \lambda_i \; &\geq \; 0 \text{ for all } i. \end{aligned} \tag{6}$$

4.1.1. *Kuhn-Tucker Conditions*

According to the Kuhn-Tucker conditions (Kuhn and Tucker, 1951), the necessary conditions for the maximum to exist are:

$$\frac{\partial L(x, \lambda)}{\partial x_j} = 0, \;\; j = 1, \ldots, n \tag{7}$$

$$\lambda_i \frac{\partial L(x, \lambda)}{\partial \lambda_i} = 0 \quad i = 1, \ldots, m \tag{8}$$

$$\lambda_i \geq 0, \quad c_i(x) - b_i \leq 0, \qquad i = 1, \ldots, m \tag{9}$$

The equations (7) and (8) form a system of $n + m$ simultaneous equations. The solutions to these equations are extreme points of (5).

4.1.2. *Multivariable Newton's method*

For $x \in \mathbb{R}^n$, the nonlinear simultaneous equations

$$F(x) = (f_1(x), \ldots, f_n(x))^T = (0, \ldots, 0)^T,$$

can be solved by the multivariable Newton's method (Burden, Faires and Reynolds, 1981), as follows

$$x^{(k+1)} = x^{(k)} - vF'(x^{(k)})^{-1} F(x^{(k)}), \tag{10}$$

where $x^{(k)}$ is the solution vector at iteration k, and v is a positive scalar. v controls the rate of convergence. If v is greater than one, the convergence is faster; if v is between zero and one, the convergence is slower. Newton's method requires the evaluation of partial derivatives of the simultaneous equations. In some applications where the functions are nonsmooth, it is convenient to use alternative approaches to the derivatives.

4.2. Branch-and-Bound Technique in Integer Programming

The branch-and-bound technique of integer programming in Reliability optimization is developed in (Nakagawa et al. , 1978) as follows:
1. Solve the problem as if all variables were real numbers. This solution is the upper bound for the maximization problem (or the lower bound for the minimization problem).
2.Choose one variable at a time that has a noninteger value, says x_j, and branch that variable to the next higher integer value for one problem and to the next lower integer value for the other. This results in two constraints $x_j \geq [x_j] + 1$ and $x_j \leq [x_j]$ that are added in the two branched problems. Solve both problems by the Lagrange multiplier method.
3. Now variable j is an integer in either branch. Fix the integers of x_j for the following steps of branch and bound. Select the branch that results in higher system reliability. Then repeat step2 on another variable $x_k \neq x_j$ for each of the new problems until all variables become integers.
4. Stop branching the problem if the solution is worse than the current best integer solution. Stop the iteration when all the desired integer variables are obtained.

4.3. Lagrange Multipliers Method, Fritz John Conditions and Multivariate Interval Newton Method

In the Fritz John conditions (Hansen, 1992) the Lagrangian function is

$$\begin{aligned} L(x,\mu,\lambda) = &= \mu\phi(x) - \sum_{i=1}^{m} \lambda_i (c_i(x) - b_i) \\ \lambda_i &\geq 0 \text{ for all } i. \end{aligned} \tag{11}$$

The nonlinear simultaneous equations $F(x) = (f_1(x), \ldots, f_n(x))^T = (0, \ldots, 0)^T$ can also be solved by using multivariate interval Newton methods, which are developed for both smooth and non smooth cases. Consider $F : \boldsymbol{x}^{(0)} \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, and suppose that $N(F; \boldsymbol{x}^{(k)}, \widehat{x}^{(k)})$ and $\boldsymbol{x}^{(k+1)}$ are defined by

$$N(F; \boldsymbol{x}^{(k)}, \widehat{x}^{(k)}) = \widehat{x}^{(k+v)},\ \ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} \bigcap N(F; \boldsymbol{x}^{(k)}, \widehat{x}^{(k)}), \tag{12}$$

where v represents the set of lower and upper bounds on the solution set of

$$F'(\boldsymbol{x}^{(k)})(x - \widehat{x}^{(k)}) = -F(\widehat{x}^{(k)}), \tag{13}$$

$\widehat{x}^{(k)}$ is the midpoint of the vector $\boldsymbol{x}^{(k)}$, and $F'(\boldsymbol{x}^{(k)})$ is an inclusion interval extension of first order of the Jacobi matrix $F'(x)$.

Convergence and existence or uniqueness verification with interval Newton methods have been studied in the past (Kearfott, 1996), (Neumaier, 1990), (Moore, 1979). In (Kearfott, 1996) the quadratic convergence of the multivariate interval Newton's method is shown.

The solution algorithm presented in this paper is applied to a sequence of intervals, beginning with some initial interval vector $\boldsymbol{x}^{(0)}$ given by the user. The initial interval can be chosen to be sufficiently large to enclose all physically feasible points. It is assumed that the global optimum will occur at an interior stationary minimum of the objective function and not at the boundaries of $\boldsymbol{x}^{(0)}$.

Interval arithmetic has been used in many optimization applications (Moore, 1979), (Ratschek and Rokne, 1990), (Neumaier, 1990), (Ratschek, 1988), (Muñoz and Pierre, 2004)

5. GlobSol

GlobSol is a package, based on interval arithmetic routines portable in Fortran 90. The definition of optimization problems is easy in GlobSol, and the algorithm configuration is flexible, see (Kearfott, 1996). GlobSol solves constrained optimization, unconstrained optimization, and nonlinear algebraic systems. It provides rigorous global search, and compiles numerous performance statistics. For constraint optimization problems, GlobSol generates a decreasing sequence of interval vectors that includes the global minimum or global maximum of the objective functions in a given feasible region $\boldsymbol{x}^{(0)}$. Nonsmooth optimization problems can be treated as smooth problems in GlobSol, see (Muñoz and Kearfott, 2004). The following solution algorithm is used in the examples of this work, and it is based on a combination of several interval global optimization techniques, for more detailed information see Chapter 5 of (Kearfott, 1996).

Solution Algorithm

Let $\boldsymbol{x}^{(0)}$ be an initial interval vector. For an interval vector $\boldsymbol{x}^{(k)}$ in the sequence of interval vectors $\boldsymbol{x}^{(0)} \supset \boldsymbol{x}^{(1)} \supset \cdots \supset \boldsymbol{x}^{(k)}$, follow the steps:

1. Compute interval evaluations for the gradient of the objective function, $\nabla C(\boldsymbol{x}^{(k)})$, and the constraint function, $R(\boldsymbol{x}^{(k)})$.

2. *Gradient range test* If the zero vector is not in the gradient of C, $0 \notin \nabla C(\boldsymbol{x}^{(k)})$, then $\boldsymbol{x}^{(k)}$ is discarded, thus no solution of $\nabla C(\boldsymbol{x}^{(k)}) = 0$ exists in this interval vector. Otherwise, the testing of $\boldsymbol{x}^{(k)}$ continues.

3. *Objective range test* Compute an interval evaluation of the objective function, $C(\boldsymbol{x}^{(k)})$. If the lower bound of $C(\boldsymbol{x}^{(k)})$ is greater than a known upper bound on the global minimum of $C(x)$, then $\boldsymbol{x}^{(k)}$ cannot contain the global minimum, and it is discarded. Otherwise, testing of $\boldsymbol{x}^{(k)}$ continues.

4. *Interval Newton test* Solve the linear interval equation system for a new interval N_k

$$C''(\boldsymbol{x}^{(k)})(N_k - \widehat{\boldsymbol{x}^{(k)}}) = -\nabla C(\widehat{\boldsymbol{x}^{(k)}}),$$

where $C''(\boldsymbol{x}^{(k)})$ is an interval evaluation of the Hessian matrix of $C(x)$, over the current interval $\boldsymbol{x}^{(k)}$, where $\widehat{\boldsymbol{x}^{(k)}}$ is the midpoint of $\boldsymbol{x}^{(k)}$. It can be shown that if $\boldsymbol{x}^{*}$ is a root of $\nabla C(\boldsymbol{x}^{(k)}) = 0$, then it is also contained in N_k.

a. If $N_k \cap \boldsymbol{x}^{(k)} = \emptyset$, then $\nabla C(\boldsymbol{x}^{(k)}) = 0$ does not have a root in $\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^{(k)}$ is discarded.

b. Evaluate $C(\widehat{\boldsymbol{x}^{(k)}})$ and find an upper bound for use in Step 3.

c. If $N_k \cap \boldsymbol{x}^{(k)} = N_k$, then there is exactly one root of $\nabla C(\boldsymbol{x}^{(k)}) = 0$ in $\boldsymbol{x}^{(k)}$, which may correspond to the global minimum.

d. If neither of the above is true, then no further conclusion can be drawn.

6. Numerical Results

The following examples compare GlobSol results with those of the Lagrange multiplier method for the unit redundancy optimization problem (6). G_i represents GlobSol results, LM represents results from the Lagrange multiplier method, and $LMBB$ results from the Lagrange multiplier method with branch-and-bound technique.

EXAMPLE 2. *A basis series consists of 4 units with costs $c_1 = 1$, $c_2 = 2$, $c_3 = 4$, and $c_4 = 8$ units of money and reliability values $p_1 = 0,2$, $p_2 = 0.4$, $p_3 = 0.6$, $p_4 = 0.8$. Design a new system configuration incorporating the unit redundancy concept to achieve an overall system reliability of $R = 0.995$ at minimum cost. Solve the dual problem maximizing the system reliability with the cost constraint.*

Considering $x = (x_1, x_2, x_3, x_4)$, an integer programming problem corresponding to this example is defined as follows:

$$\textit{Minimize } \phi(x) = x_1 + 2x_2 + 4x_3 + 8x_4$$

subject to the constraints

$$(1 - 0.8^{x_1})(1 - 0.6^{x_2})(1 - 0.4^{x_3})(1 - 0.2^{x_4}) \geq 0.995$$

$$x_i \textit{ positive integer for } i = 1, 2, 3, 4.$$

The following F90 file is used to solve this optimization problem with GlobSol.

```
PROGRAM RELIABILITY
USE CODELIST-CREATION
PARAMETER (NN=4)
TYPE(CDLVAR), DIMENSION(NN):: X
TYPE(CDLLHS), DIMENSION(1):: PHI
TYPE(CDLINEQ), DIMENSION(1):: G
TYPE(CDLEQ), DIMENSION(4):: C
TYPE(INTERVAL) :: IPI
INTEGER I
IPI = IVL(3.14159265358979323846264338327D0)
CALL INITIALIZE-CODELIST(X)
PHI(1) = X(1) + 2*X(2) + 4*X(3) + 8*X(4)
G(1) = -(1-0.8D0**X(1))*(1-0.6D0**X(2))*(1-0.4D0**X(3))*(1-0.2D0**X(4))
+ 0.995D0
DO I = 1,4
C(I) = SIN(IPI*X(I))
END DO
CALL FINISH-CODELIST
END PROGRAM RELIABILITY
```

Exerts from the output file

Output from FIND-GLOBAL-MIN on 02/09/2005 at 13:06:25.

Version for the system is: November 22, 2003
Box data file name is: reliability.DT1
Initial box:

$$[0.2000D + 01, 0.5050D + 02] \quad [0.2000D + 01, 0.5050D + 02]$$

$[0.2000D + 01, 0.5050D + 02]$ $[0.2000D + 01, 0.5050D + 02]$ *(lines deleted)*

LIST OF BOXES CONTAINING VERIFIED FEASIBLE POINTS:

Box no.: 1
Box coordinates:
$[0.3099D + 02, 0.3301D + 02]$ $[0.1400D + 02, 0.1400D + 02]$
$[0.7000D + 01, 0.7000D + 01]$ $[0.4000D + 01, 0.4000D + 01]$
PHI: $[0.1190D + 03, 0.1210D + 03]$
(lines deleted)
Box contains the following approximate root:

Table I. Results for Example 2

Method	Solution	Cost	Reliability
G_1	$(28, 14, 8, 4)$	120	.99504
G_2	$(30, 15, 7, 4)$	120	.99506
G_3	$(32, 14, 7, 4)$	120	.99519
G_4	$(30, 13, 8, 4)$	120	.99521
LM	$(30, 14, 8, 4)$	122	.99573

0.3200D+02 0.1400D+02 0.7000D+01 0.4000D+01

OBJECTIVE ENCLOSURE AT APPROXIMATE ROOT:
$[0.1200D + 03, 0.1200D + 03]$
(lines deleted)

Box no.: 2
Box coordinates:
$[0.2905D + 02, 0.3095D + 02]$ $[0.1300D + 02, 0.1300D + 02]$
$[0.8000D + 01, 0.8000D + 01]$ $[0.4000D + 01, 0.4000D + 01]$
PHI: $[0.1191D + 03, 0.1209D + 03]$
Box contains the following approximate root:
0.3000D+02 0.1300D+02 0.8000D+01 0.4000D+01

OBJECTIVE ENCLOSURE AT APPROXIMATE ROOT:
$[0.1200D + 03, 0.1200D + 03]$

ALGORITHM COMPLETED WITH LESS THAN THE MAXIMUM NUMBER, 20000 OF BOXES.
Number of bisections: 286
No. dense interval residual evaluations – gradient code list: 4292
Total number dense interval constraint gradient component evaluations: 48208
Total number dense point constraint gradient component evaluations: 312
(lines deleted)
Number Fritz-John matrix evaluations: 383
Total number of boxes processed in loop: 360
BEST-ESTIMATE: 0.1200D+03
Overall CPU time: 0.1054D+03

The original system cost is, $C_0 = 15$, the minimum cost for the unit redundancy concept by using GlobSol is $C = 120 = 8C_0$, and by using LM is $C = 122 = 8.13C_0$ (see Table I). The dual optimization problem, to maximize the overall system reliability with a maximum cost of 120, was also solved with GlobSol obtaining similar results. In both problems the optimum solution was $(x_1, x_2, x_3, x_4) = (30, 13, 8, 4)$, with a total cost of $C = 120$ and a maximum system reliability of $R = .9952$. when the cost constraint is changed to a maximum cost of 123, the optimum solution obtained with GlobSol and LM is the same $(x_1, x_2, x_3, x_4) = (30, 14, 8, 4)$, with a total cost of $C = 122$ and a maximum system reliability of $R = .9957$.

EXAMPLE 3. *Similar to Example 2, with unit costs $c_1 = 1$, $c_2 = 2$, and $c_3 = 3$.*

Table II. Results for Example 4

Method	Solution	Cost	Reliability
G_1	$(13, 8, 6)$	47	.990718
G_2	$(11, 9, 6)$	47	.990353
LM	$(12, 9, 6)$	48	.991795

The original system cost is $C_0 = 6$, the minimum cost for the unit redundancy concept by using GlobSol is $C = 47 = 7.83C_0$ and by using LM is $C = 48 = 8C_0$ (see Table II). The dual optimization problem, to maximize the overall system reliability with a maximum cost of 47, was also solved with GlobSol obtaining similar results. In both problems the optimum solution was $(x_1, x_2, x_3) = (13, 8, 6)$, with a total cost of $C = 47$ and a maximum system reliability of $R = .990718$.

EXAMPLE 4. *Similar to Example 1, with unit costs $c_1 = 2$, $c_2 = 3$, $c_3 = 4$, and $c_4 = 5$, and reliability values $p_1 = p_2 = p_3 = p_4 = 0.5$. Design a new system configuration incorporating the unit redundancy concept to achieve an overall system reliability of $R = 0.98$ at minimum cost.*

Table III. Results for Example 4

Method	Solution	Cost	Reliability
G_1	$(8, 8, 8, 7)$	107	.980606
LM	$(9, 8, 8, 7)$	109	.982528

The original system cost is $C_0 = 14$, the minimum cost for the unit redundancy concept by using GlobSol is $C = 107 = 7.64C_0$ and by using LM is $C = 109 = 7.79C_0$ (see Table III). The dual optimization problem, to maximize the overall system reliability with a maximum cost of 107, was also solved with GlobSol obtaining similar results. In both problems the optimum solution was $(x_1, x_2, x_3, x_4) = (8, 8, 8, 4)$, with a total cost of $C = 107$ and a maximum system reliability of $R = .980606$.

Table IV. Data for Example 5

Stage, j	1	2	3	4
r_j	0.80	0.70	0.75	0.85
c_{1j}	1.2	2.3	3.4	4.5
c_{2j}	5	4	8	7
$b_1 = 56$				
$b_2 = 120$				

EXAMPLE 5. *A 4-stage series system with two linear constraints is formulated as a pure integer programming problem. The decision variables,* $x = (x_1, x_2, x_3, x_4)$*, are the number of redundancies at each stage. The problem is formulated as follows*

$$\begin{aligned} \textit{maximize } R &= \prod_{i=1}^{4}[1-(1-r_i)^{x_i}] \\ \textit{subject to } \sum_{j=1}^{4} c_{ij}x_i &\leq b_i, \quad i = 1,2 \end{aligned} \tag{14}$$

With the data given in Table IV, the real solution obtained by the LM and the Kuhn-tucker conditions is, $x = (5.11672, 6.30536, 5, 23536, 3.90151)$*, using interval and LMBB techniques give the same integer solution* $x = (5, 6, 5, 4)$*. Even both methods provide the same conclusions about the decision variables, interval techniques provide a more rigorous reasoning by guaranteeing the optimality for this problem.*

EXAMPLE 6. *A 5-stage series system with three nonlinear constraints is formulated as a mixed integer programming problem. Both the number of redundancies,* x_j*, and the component reliability,* r_j*, are to be determined. The problem from (*Tillman et al., 1985*) is*

$$\begin{aligned} \textit{maximize } R_s(x,r) &= \prod_{i=1}^{4}[1-(1-r_i)^{x_i}] \\ \textit{subject to } \; g_1(x) &= \sum_{j=1}^{5} p_j x_j^2 - P \leq 0 \\ g_2(x,r) &= \sum_{j=1}^{5} \alpha_j (\frac{-t}{\ln r_j})^{\beta_j}(x_j + \exp(x_j/4)) - C \leq 0 \\ g_3(x) &= \sum_{j=1}^{5} \omega_j x_j \exp(x_j/4)) - W \leq 0 \end{aligned} \tag{15}$$

$x_j \geq 1$ *are integers;* $\quad 0 < r_j < 1$*; for all j.*

*With the data given in Table V, the problem was solved with the methods: the LMBB, and with a combination of the sequential method, Hooke and Jeeve Pattern Search, and the heuristic redundancy allocation method HJHRA (*Tillman et al., 1978*).*

The results summarized in Table VI show that the LMBB method with the solution $(R_s, r, x) = (.9298, .7796, .8007, .9023, .7104, .8595, 3, 3, 2, 3, 2)$ *is superior to the HJHRA method with the solution* $(R_s, r, x) = (.9149, .7582, .8000, .9000, .8000, .7500, 3, 3, 2, 2, 3)$ *given in (*Tillman et al., 1978*). This mixed integer programming problem has many local optima. The HJHRA method has the drawback of being trapped by a local optimum, and the LMBB method overcomes this drawback and it is quite effective. Interval techniques provide the optimal solution for the redundancy allocation problem related to this problem. We could not verify the solution provided by the LMBB method to the mixed integer programming problem by using interval techniques in our computer systems.*

Table V. Data for Example 6

j	α_j	p_j	ω_j	P	C	W
1	2.33×10^{-5}	1	7			
2	1.45×10^{-5}	2	8			
3	5.41×10^{-5}	3	8	110	175	200
4	8.05×10^{-5}	4	6			
5	1.95×10^{-5}	2	9			
	$\beta_j = 1.5, \quad j = 1, \ldots, 5$		t=1000			

Table VI. Comparison of Methods

	LMBB	HJHRA
Number of redundancies	$x = (3, 3, 2, 3, 2)$	$x = (3, 3, 2, 2, 3)$
Component reliability	$r = (.7796, .8007, .9023, .7104, .8595)$	$r = (.7582, .8000, .9000, .8000, .7500)$
System reliability	$R_s = .9298$	$R_s = .9149$
Slack of g_1	27	28
Slack of g_2	0.00001	0.033727
Slack of g_3	10.57248	1.4118

7. Conclusions

Interval arithmetic techniques, proved to be an effective tool to determine optimal design configurations for systems with unit redundancy, and can be used to solve reliability-redundancy allocation problems. Most of the results were obtained in 0.1302D+01 CPU seconds, and they are quite convincing when compared with those obtained in (Way et al., 1987) with LMBB technique that involved 3 CPU seconds. Interval arithmetic techniques are competitive alternatives since they provide management with different options and flexibility.

Acknowledgements

The authors wish to thank OSP-SUBR for its financial support, to the organizers of the REC 2006 for accepting this paper, and to Dr. Baker Kearfott for his technical support with GlobSol.

References

Agrafiotis, G. K. and M. Z. Tsuokalas. Reliability Analysis and Optimization Applications of a Two-Unit Standby Redundant System with Spare Units. *Microelectron. Reliab.*, 34(9):1469–1475, 1994.

Bulfin, R. L. and C. Y. Liu. Optimal Allocation of Redundant Components for Large Systems. *IEEE transactions on Reliability*, R-34:241–247, 1985.

Burden, R. L., J. D. Faires, and A. C. Reynolds. *Numerical Analysis.* Prindle, Weber and Schmidt, 2nd edition, 1981.

Coit, D. W. Cold-Standby Redundancy Optimization for non Repairable Systems. *IIE Transactions*, 33:471–478, 2001.

Gurov, S. V., L. V. Utkin, and I. B. Shubinsky. Optimal Reliability allocation of Redundant Units and Repair Facilities by arbitrary Failure and Repair Distributions. *Microelectron, Reliab.*, 35(12):145–1460, 1995.

Hansen, E. R. *Global Optimization Using Interval Analysis.* Marcel Dekker, Inc., New York, 1992.

Harunuzzaman, M. and T. Aldemir. Optimization of Standby System Maintenance Schedules in Nuclear Power Plants. *Nuclear technology*, 113(3):354–367, 1996.

Kearfott, R. B. *Rigorous Global Search: Continuous Problems.* Kluwer Academic Publishers, Netherlands, 1996.

Kuhn, N. W. and A. W. Tucker. Nonlinear Programming. *Proc. second Berkeley Symp. Mathematical Statistics and Probability*, Berkeley, California, 481–492, 1951.

Misra, K. B. and M. D. Ljubojevic. Optimal Reliability design of a system: A new look. *IEEE transactions on Reliability*, R-22:255–258, 1973.

Moore, R. E. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, 1979.

Moore, R. E.: *Interval Arithmetic and Automatic Error Analysis in Digital Computing.* PhD thesis, Stanford University, October 1962.

Muñoz, H. Interval Slopes and Twin Slope Arithmetic in Nonsmooth Optimization. Ph.D. thesis, University of Louisiana at Lafayette, 2001.

Muñoz, H. and B. Kearfott. Slope Intervals, Generalized Gradients, Semigradients, Slant Derivatives and Csets. *Reliable Computing*, 10(3):2004.

Muñoz, H. and E. Pierre. Interval Arithmetic Optimization Technique for System Reliability with Redundancy. *Proceedings of the 8th International Conference on Probabilistic Methods Applied to Power Systems*, 2004.

Nakagawa, Y., K. Nakashima, and Y. Hattori. Optimal Reliability allocation by branch-and-bound technique. *IEEE transactions on Reliability*, R-27:31–38, 1978.

Neumaier, A. *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, England, 1990.

Ramakumar, R. *Engineering Reliability, Fundamentals and Applications.* Prentice Hall, New York, 1993.

Ratschek, H. Some Recent Aspects of Interval Algorithms for Global Optimizations. *Reliability in Computing 'The Role of Interval Methods in Scientific Computing'.* Academic Press, pp. 325–339, 1988.

Ratschek, H. and J. Rokne. Interval Tools for Global Optimization. *Computers Math. Applic.*, 1:1–11, 1990.

Tillman, F. A., C. L. Hwang, and W. Kuo. Optimization Techniques for System Reliability with Redundancy–A Review. *IEEE Transactions on Reliability*, R-26(3):148–155, 1978.

Tillman, F. A., C. L. Hwang, and W. Kuo. Optimization of Systems Reliability. Marcel Dekker, 1985.

Way, K., L. Hsin-Hui, X. Zhongkai, and Z. Weixing. Reliability Optimization with the Lagrange-Multiplier and Branch-and-Bound Technique. *IEEE Transactions on Reliability*, R-36(5):624–630, 1987.