

Optimal Allocation for Reliability Analysis of Series Parallel System using Dynamic Programming

¹D. Ravi Kumar & ²V. Sankar

^{1&2} Member IEEE VNR Vignana Jyothi Institute of Engg. & Tech., Hyderabad, India
JNTUACEA, Anantapur, India
E-mail: ravi44d@gmail.com¹, sankarvelamury@ieee.org²

Abstract - In general, optimal allocations for Series-Parallel redundant configurations being carried out using unit or component redundancies. However, Dynamic programming approach by which the optimal allocation can be used when the values of reliabilities and cost for each component are known. The advantage of Dynamic programming approach is that it is simple and it requires less processing time.

In this paper, the optimal allocation for series parallel Reliability Logic Diagram of a system is considered. The deterministic Dynamic Programming approach for Reliability optimization has been used and optimal allocation is obtained with different number of components at each stage. Whereas, in the earlier methods, either unit or component redundancies are only used for estimating Optimum reliability.

Keywords - *Optimal allocation; Reliability; Dynamic Programming; Redundancy Optimization.*

I. INTRODUCTION

Optimal solutions to the redundancy allocation problem [1] are determined for systems designed with multiple k-out-of-n subsystems in series. The objective is to select the components and redundancy levels to maximize system reliability given system-level constraints. The individual subsystems may use either active or cold-standby redundancy, or they may require no redundancy.

During the design phase of a product, reliability engineers are called upon to evaluate the reliability of the system. The question of how to meet a reliability goal for the system arises when the estimated reliability is inadequate. This then becomes a reliability allocation problem at the component level which is achieved by formulating cost function using Nonlinear programming [2].

A mathematical model [3] of a redundancy allocation problem with mixing components redundant in subsystems of a series-parallel system when the redundancy strategy can be chosen for individual subsystems. In practice both active and cold standby redundancies are used within a particular system design, and the choice of the redundancy strategy becomes an

additional decision variable. The model selects the best redundancy strategy, combination of components, and levels of redundancy for each subsystem in order to maximize the system reliability under system level constraints.

In this paper, the optimum allocation of subsystems for a series system to achieve maximum reliability without violating the constraints is considered. The problem of redundancy allocation has been studied in great detail for different formulations and optimization algorithms [4]. The majority of them are however limited in their application to simple systems consisting of exponential distribution [5] of the components. The problem is to be solved by Dynamic Programming approach. Dynamic programming can be used for problems where the value of reliability of components of the system is deterministic.

Unit and Component redundancy techniques [6] are considered for the optimum allocation of components.

II. DYNAMIC PROGRAMMING

The heart of the Dynamic Programming approach is the principle of optimality set forth by Bellman [4]. It states that an optimal policy has the property that

whatever the initial stage and the initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from first decision.

A. Characteristics of Dynamic Programming problems

The basic features [4] which characterize dynamic programming problems are presented and discussed below.

1. The problem can be divided into stages, with a policy decision required at each stage.
2. Each stage has a number of states associated with it.
3. The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage according to the probability distribution.
4. Given the current state, an optimal policy for the remaining stages is independent of the policy adopted in previous stages.
5. The solution procedure begins by finding the optimal policy for each state of last stage.
6. A recursive relationship that identifies optimal policy for each state at stage n , given the optimal policy for each state at stage $(n+1)$, is available.

For the above problem, this recursive relationship is [4]

$$f_n^*(s) = \min_{x_n} \{C_{sxn} + f_{n+1}^*(x_n)\} \quad (1)$$

Where $f_n^*(s)$ is the optimal policy with starting state s at stage n , which requires finding the minimum value of x_n . The policy would consist of using this value of x_n and then following the optimal policy when starting in state x_n at stage $(n+1)$, $(f_{n+1}^*(x_n))$.

The above precise form of the recursive relationship differs somewhat among dynamic programming problems [4]. Thus, let the variable (or vector) x_n be the decision variable at stage n ($n=1, 2, \dots, N$). Let $f(s, x_n)$ for this state 's' at stage 'n' and x_n is selected. Let $f_n^*(s)$ be the maximum/ minimum value of $f_n(s, x_n)$ over all possible values of x_n . The recursive relationship will always be of the form

$$f_n^*(s) = \max_{x_n} / \min_{x_n} \{f_n(s, x_n)\} \quad (2)$$

Where $f_n(s, x_n)$ would be written in terms of $s, x_n, f_{n+1}^*(.)$ and will be a measure of the first stage effectiveness/ineffectiveness of x_n . Using the Eqn. (1), the solution procedure moves backward stage by stage, each time finding the optimal policy for each state of that stage, until it finds the optimal policy when starting

at the initial stage. For all dynamic programming problems, a table such as the following would be obtained for each stage ($n = N, N-1, \dots, 1$):

TABLE I. Dynamic Programming Data

s	$f_n^*(s)$	x_n^*

When this table is finally obtained for the initial stage ($n=1$), the problem of using Eqn. (2) will be solved. Since the initial state would be known, the initial decision will be specified by x_1^* in this table. The optimal value for the other decision variable would then be specified by the other table according to the state of the system at those stages.

III. DETERMINISTIC DYNAMIC PROGRAMMING

In this section, the dynamic programming approach to deterministic probability problems, where the state at the next stage is to be completely evaluated by the state and policy decision at the current stage is described.

Thus, at stage ' n ', the process will be in some state s_n . Making policy decision x_n then moves the process to some state s_{n+1} . From that point onwards, the objective function value for optimal policy $f_{n+1}^*(s_{n+1})$ is to be calculated. The policy decision x_n also makes some contribution to the objective function. Combining these two quantities in an appropriate way provides the objective function value $f_n(s_n, x_n)$ beginning at stage 'n'. Minimizing with respect to x_n , then gives $f_n^*(s_n) = f_n(s_n, x_n)$. After doing this for each possible value of s_n , the solution procedure is ready to move back to one stage.

One way of categorizing deterministic dynamic programming problem is by the form of objective function. For example, the objective might be to minimize the sum of contribution from the individual stages or to minimize a product of such terms, and so on. Another categorization is in terms of the nature of the set of states for the respective stages. In particular, the state s_n might be representable by a discrete state variable or by a continuous state variable or a state vector (if more than one variable) is required.

A. Problem formulation

Consider electrical equipment which has 'n' number of components 1, 2, 3, ..., n which are connected in series (i.e., if one component fails then the equipment does not work). To improve the reliability of the equipment each component is to be supplemented by one or more parallel units (In case the component fails then the supplementary unit works and consequently the system remains working). It is required that the data of

cost of parallel units and reliability at each component be known a priori.

The constraints under which the problem solved are :

1. The maximum resources allocated for the specified system cost.
2. The maximum allowable parallel subsystems that can be allowed to a system are specified.

The aim is to find the following:

- To find the optimal allocation of redundancy for the given problem
- To find the number components required at each stage.
- To find the optimal redundancy of each component for achieving reliability goal using component wise redundancy with dynamic programming.

B. Computational Procedure

To achieve the first aim, the following algorithm is developed.

Firstly, the problem is solved for the first stage.

Step 1: For each value of cost(s), from minimum required cost for that stage to the maximum allowable cost for that stage, the objective function (f_n) is to be calculated for each value of allowable number of parallel stages, using Eqn. (3).

$$f_n(x) = p(x_n) \quad (3)$$

Step 2: For each value of cost, the maximum value of objective function (f_n^*) and the value of number of parallel elements for which the function to be maximized (x_n^*) is to be calculated using Eqn. (4).

$$f_n^*(x) = \max f_n(x) \quad (4)$$

Now, the problem is to be solved from second to the penultimate stage as follows:

Step 3: For each value of cost(s), from minimum required cost for that stage to the maximum allowable cost for that stage, the objective function (f_n) is to be calculated for each value of allowable number of parallel stages. Here, the Eqn. (3) is changed as:

$$f_n(x) = p(x_n) * f_{n-1}^*(x) \quad (5)$$

Step 4: For each value of cost, the maximum value of objective function (f_n^*) and the value of number of parallel elements for which the function is maximized (x_n^*) is to be calculated using Eqn. (6).

$$f_n^*(x) = \max f_n(x) \quad (6)$$

Step 5: The Steps 3 and 4 are repeated but the cost varies from minimum to the maximum costs.

Step 6: The final values of x_n^* are the optimal allocation of reliability. The results are then tabulated.

Next, it is proved that component wise redundancy is better than unit redundancy, the reliability values with two different configurations are considered. They are

1. Two units of the system with optimal configuration are arranged in parallel and the reliability value is calculated.
2. Two units of each component are arranged in parallel and the reliability of the whole system is calculated.

These two values are compared and the configuration with higher reliability is termed as the better configuration.

To find the number of parallel units required to achieve the reliability goal, the number of parallel units are increased one by one till the desired reliability is achieved.

Similarly, to find the optimal allocation by component wise redundancy, the following procedure is followed.

Let the optimal allocation for the given problem be n_1, n_2, n_3, \dots , parallel components in stage 1, 2, 3, ... respectively. Then n_1 parallel components of stage 1 are considered as a single unit. Similarly, n_2 components of stage 2 are considered as single unit. Similarly, for the n stages, the optimal number of components for each stage is supposed to be one unit. To achieve the reliability goal, without any cost constraint, dynamic programming is applied till the total system reliability equals desired reliability. Algorithm has implemented using MATLAB and the results are presented in the next section.

III. RESULTS

A. Results for optimal allocation of redundancy for the given system

Enter the number of stages in the system: 4

Enter the maximum allowable Stage cost (maxcost) in Rupees: 20

Enter the maximum number of parallel components allowable: 4

Enter the values of reliabilities and cost in Rupees of each stage in matrix form

[0.6 2; 0.7 4; 0.8 6; 0.9 8]

Enter the values of reliabilities and costs of each stage in matrix form

[0.4 1; 0.5 3; 0.7 5; 0.8 7]

Enter the values of reliabilities and costs of each stage in matrix form

[0.3 1; 0.4 2; 0.5 3; 0.6 4]

Enter the values of reliabilities and costs of each stage in matrix form

[0.4 2; 0.5 3; 0.6 4; 0.8 6]

1) *Optimal allocation of reliability by dynamic programming*

The optimal number of parallel components and costs for each stage are as follows:

The optimum number of parallel components for stage 4 = 3

The cost of stage 4 = 4

The reliability of stage 4 = 0.6

The optimum number of parallel components for stage 3 = 4

The cost of stage 3 = 4

The reliability of stage 3 = 0.6

The optimum number of parallel components for stage 2 = 3

The cost of stage 2 = 5

The reliability of stage 2 = 0.7

The optimum number of parallel components for stage 1 = 1

The cost of stage 1 = 2

The reliability of stage 1 = 0.6

The maximum reliability achieved is 0.1512.

Results to find the number of redundant units to achieve reliability goal is as follows:

B. *Improvement of Reliability by unit redundancy*

Values of Reliabilities and costs with unit redundancy are given in Table 2.

Table II. Reliability and cost values

State No.	Reliability of each Stage	Total Cost in Rupees
1	0.151200	20
2	0.279539	40
3	0.388472	60
4	0.480935	80
5	0.559418	100
6	0.626034	120

State No.	Reliability of each Stage	Total Cost in Rupees
7	0.682578	140
8	0.730572	160
9	0.771309	180
10	0.805870	200
11	0.835237	220
12	0.860149	240
13	0.881295	260
14	0.899243	280
15	0.914477	300

The number of parallel stages for achieving a minimum reliability goal of 0.9 by unit redundancy is 15.

Results to find the optimal allocation of components to achieve reliability goal are as follows:

C. *Improvement of Reliability using component redundancy*

The optimal number of parallel components for each stage are obtained as follows:

Stage 4 = 3

Stage 3 = 1

Stage 2 = 1

Stage 1 = 1

The reliability achieved = 0.235872 and the total number of components is 6 and the total cost is 120, whereas the total cost with unit redundancy will be 300.

(a) **For maxcost = 40**

The optimal number of parallel components for each stage are obtained as follows:

Stage 4 = 3

Stage 3 = 2

Stage 2 = 3

Stage 1 = 3

The reliability achieved = 0.716051 and the total number of components is 11 only with a total cost of Rs. 440, whereas the total cost with unit redundancy will be Rs. 600.

(b) **For maxcost = Rs. 60**

The optimal number of parallel components for each stage are obtained as follows:

Stage 4 = 4

Stage 3 = 4

Stage 2 = 4

Stage 1 = 4

The reliability achieved = 0.917656 and the total number of components is 16 only with a total cost of Rs. 640 only, whereas the total cost with unit redundancy will be Rs. 900.

D. PLOTS

Plots are drawn to show the relation between various parameters. As the reliability goal increases the cost of the components required to achieve it also increases exponentially. This is shown in Fig. 1 in the form of a graph between the various reliability values and the cost of achieving it.

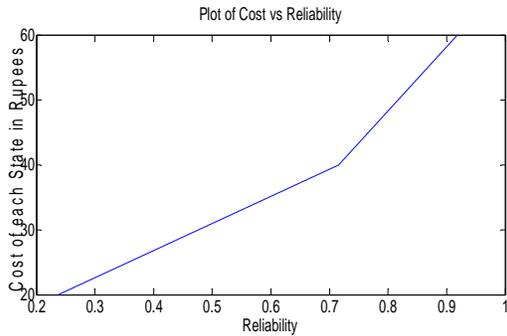


Fig. 1 : Plot of Cost vs. Reliability

The bar graph is shown in Fig. 2 with costs Vs number of states that are to be required in each stage for a reliability of 0.9 with unit redundancy configuration.

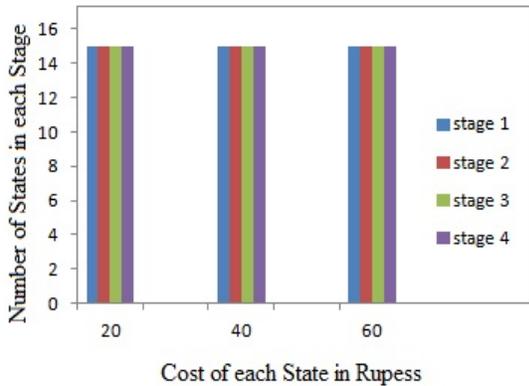


Fig. 2 : Plot of number of States in each stage for various costs using unit redundancy

The optimum number of states in each stage for different costs do not follow a specific pattern. It is shown by bar graph in Fig.3. This is obtained using component redundancy at each stage.

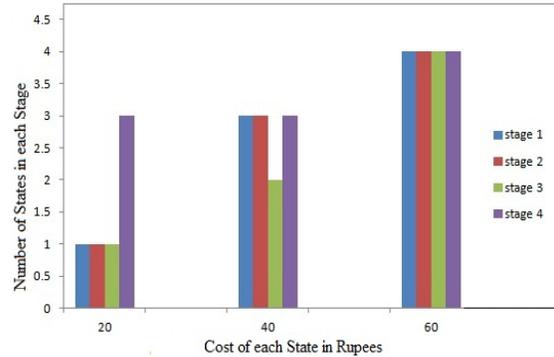


Fig. 3 : Plot of Optimum number of states in each stage using Component redundancy

The costs of achieving 0.9 minimum reliability goal using unit and component redundancies respectively, are shown in Fig. 4 for the state cost of Rs. 20.

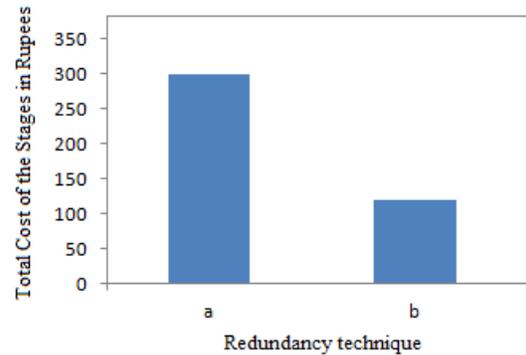


Fig. 4: Comparison of allocation by unit and component redundancy

In Fig. 4, ‘a’ stands for reliability allocation by unit redundancy and ‘b’ denotes the allocation by component wise redundancy. From Fig. 4, it can be observed that the component redundancy is better than unit redundancy.

VI. CONCLUSIONS

Dynamic programming is very useful technique for making a sequence of interrelated decisions. It requires formulating an appropriate recursive relationship for each individual problem. However, it provides a great computational saving over using exhaustive enumeration to find the combination of decision, especially for large problems. The number of redundant units required for achieving given reliability goal is also determined. Finally, optimal allocation of component redundancy to achieve the reliability goal is also determined by dynamic programming, and the advantage of component redundancy over unit redundancy is presented.

REFERENCES:

- [1] David W. Coit and Jiachen Liu, "System Reliability Optimization with k-out-of-n subsystems", *International Journal of Reliability, Quality and Safety engineering*, Vol. 7, No. 2, 2000, 129-142.
- [2] A. Mettas, Reliability allocation and optimization for complex systems. *Proc. of Annual Reliability and Maintainability Symposium*, Los Angeles, Jan. 2000, pp. 216-221.
- [3] R.Tavakkoli-Moghaddam, "A New mathematical Model for a Redundancy Allocation Problem with Mixing Components Redundant and Choice of Redundancy Strategies", *Applied mathematical Sciences*, Vol. 1, 2007, No. 45, pp. 2221 – 2230.
- [4] F.A. Tillman, C.V. Hwang & W. Kuo, *Optimization of Systems Reliability*, Cambridge University Press, 2001.
- [5] R. Billinton, R. N. Allan, *Reliability Evaluation of Engineering Systems*, BS Publications, 2007.
- [6] E. Balagurusamy, *Reliability Engineering*, Tata Mc Graw-Hill Publishing Company Limited, 2010.

