Invariants in Hierarchical-System Optimization for Reliability and Maintainability

MENG-LAI YIN* and JOHN ANGUS²

1Electrical & Computer Engineering, California State Polytechnic University, Pomona
2School of Mathematical Sciences, Claremont Graduate University, USA

(Received on August 9, 2010, revised on October 11, 2010)

Abstract: The “invariants” in a process are the non-changing parts. In this paper, invariants in determining the redundancy allocation to optimize system reliability and maintainability are exploited. This article demonstrates how recognizing the computational invariants can lead to efficient system assessments.

Keywords: Optimization, redundancy allocation, reliability, series MTBF.

1. Introduction

Optimal reliability design problems are NP-hard [1]. Efficient heuristic methods intended for redundancy allocation are always desirable. Here, the purpose of redundancy allocation is to optimize system reliability and maintainability, the latter measured by series MTBF (Mean Time between Failures), as described next. The notations are:

- \( C_0 \) basic, minimally functional system
- \( C_i \) a system configuration derived from \( C_0 \) by adding redundancy
- \( R_{sys}(C_i) \) system reliability for configuration \( C_i \), or simply reliability
- \( \theta_{sys}(C_i) \) system series MTBF for configuration \( C_i \)
- \( \delta \) system reliability requirement
- \( \omega \) system series MTBF requirement
- \( n \) number of critical components in the system
- \( x_s \) number of redundant components for critical component \( s \), \( 1 \leq s \leq n \)
- \( \lambda_s \) failure rate for critical components \( s \), \( 1 \leq s \leq n \) (assumed constant)

The system being considered requires \( n \) critical components (subsystems) to operate; redundant components can be added to improve reliability. For example, a critical component \( s \) can have \( x_s > 0 \) for a total of \( (x_s+1) \) components of type \( s \). Thus, each of the \( n \) critical subsystems forms a one-out-of-(\( x_s+1 \)) parallel subsystem. The present reliability analysis is restricted to hierarchical systems, applying a series-parallel hierarchical decomposition method [2], [3]. The series MTBF is a measure of how frequently the system must be repaired. It is defined as

\[
\frac{1}{\sum_{x \in S} (x + 1) \times \lambda_x}.
\]

Although the discussion is focused on redundancy allocation, the technique of invariants can be applied to other applications employing exhaustive, heuristic, or meta-heuristic approaches based on hierarchical system structures.

Regarding the problem considered here, the two attributes, i.e., reliability and series MTBF, are antithetical. That is, adding redundancy improves reliability, but degrades (i.e.

---

*Corresponding author’s email: myin@csupomona.edu

198
decreases) the series MTBF; while reducing redundancy improves series MTBF but degrades (i.e. decreases) system reliability. Hence, two problems are relevant:

(P1) Maximize $\theta_{\text{sys}}(C_i)$; Subject to $R_{\text{sys}}(C_i) \geq \delta$

(P2) Maximize $R_{\text{sys}}(C_i)$; Subject to $\theta_{\text{sys}}(C_i) \geq \omega$

The technique we employ is inspired by the neocortex model [4] where it is stated that a major source of efficiency in human brain is the use of invariants in making predictions.

2. The Process

Given an initial configuration $C_0$, P1 is to maximize the series MTBF while meeting the reliability requirement; P2 is to maximize the reliability with series MTBF bounded. As depicted in Figure 1(a), for P1 to be meaningful, the initial configuration $C_0$ must be in area II, where $R_{\text{sys}}(C_0) \geq \delta$ as increasing series MTBF can only degrade system reliability. Similarly, to pursue P2, $\theta_{\text{sys}}(C_0) \geq \omega$ is necessary. The goal is to reach a configuration $C_i$ in area I where both requirements are met. Note that this is not always feasible, e.g., the initial configuration may fall in area III, or the process may end at the boundary of area II and III (or III and IV) due to design constraints or components limitations. These circumstances show that limits are reached and the ideal configuration is not possible.

We follow the strategy proposed by Barlow and Proschan [5], which asserts that the optimization problem can be solved by a series of allocations. As illustrated in Figure 1(b), the iterative process starts from an initial (or updated) configuration. The process determines whether the optimization goal has been achieved, or design limits are reached. If there is still room for improvement, the effects of all potential candidates will be evaluated, and the component with the best performance will be selected, which leads to the next configuration.

3. The Invariants

In the above process, the most time-consuming part is the assessment of all candidates’ effects on the system’s reliability and series MTBF. During each transition, note that only one component’s redundancy status will be changed. This implies that the majority of components are not changing and the corresponding computations don’t need to be recalculated. For example, a configuration $C_i$ has $n$ critical components, where the failure rate for component $s$ is denoted as $\lambda_s$, $1 \leq s \leq n$. The system’s series MTBF $\theta_{\text{sys}}(C_i)$ is then computed as

$$\theta_{\text{sys}}(C_i) = \frac{1}{\left(\sum_{s=1}^{n} (x_s + 1) \times \lambda_s\right)}$$

\hspace{1cm} (1)

Figure 1: Redundancy Allocation Problems and Process
To compute $\theta_{sys}(C_i)$ using (1) takes $O(n)$ time. However, since only one redundant component will be added (P2) or removed (P1) during each transition, calculating the series MTBF can reuse the previous results, as illustrated in (2) and (3). Thus, calculating $\theta_{sys}(C_{i\pm 1})$ during one transition only takes $O(1)$ time.

\[
P1: \quad \theta_{sys}(C_{i+1}) = \lambda_i \left(1 - \frac{1}{\theta_{sys}(C_i)} + \lambda_i\right) \quad \text{(2), or}
\]

\[
P2: \quad \theta_{sys}(C_{i-1}) = \lambda_i \left(1 - \frac{1}{\theta_{sys}(C_i)} + \lambda_i\right) \quad \text{(3)}
\]

The invariants in system reliability assessment are illustrated using the example shown in Figure 2, where the system contains three sub-systems, and each sub-system is further decomposed into lower components. Since only one component’s redundancy status will be changing, only the nodes on the affected path need to update their reliabilities, leaving the majority of nodes in the tree unaffected. The data associated with the nodes on the unaffected paths are the invariants.

**Procedure bottom-up-Rel(node)**

\[
\begin{align*}
R_{\text{node}} &= R_{\text{comp}}; \\
\text{do} & \quad \{ \\
R_j &= R_{\text{node}}; \\
\text{Re-assess } R_j \text{ using equation (4)} \\
\text{if node’s redundancy }> 0 & \{ \\
R_{\text{node}} &= 1 - (1 - R_{\text{node}})^{x+1}; \\
\text{node} &= \text{node->parent}; \\
\} \\
\text{while node is not the root} & \}
\end{align*}
\]

**Figure 2:** A Hierarchical System Example and the Bottom-Up Reliability Estimation

Unlike the traditional way of assessing system’s reliability where the entire tree is traversed in a top-down manner, a bottom-up procedure starting from the changing node is proposed (shown in Figure 2). This procedure traces along the affected path back to the root, hence provides a more efficient way of assessing the system reliability. For an affected series node (with $t$ blocks), if block $j$ is the changing node, its reliability $R_{\text{node}}$ can be re-calculated as

\[
R_{\text{node}} = R_j \times \prod_{i=j+1}^{t} R_i \quad \text{(4)}
\]

The number of nodes visited in the affected path corresponds to the height of the tree. Denoting the number of stages for each node as $m$, and the height of the tree structure as $h$, the complexity of the bottom-up procedure is of order $O(mh)$, compared to the complexity of the top-down procedure which is $O(m^h)$!

**References**


