A note on "Optimal resource allocation for security in reliability systems"

Tonguç Ünlüyurt[a] and Endre Boros[b]

[a] *Sabanci University, Orhanli, Tuzla, 34956, Istanbul, Turkey*

[b] *RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ, 08854, USA*

Abstract

In a recent paper by Azaiez and Bier [Optimal resource allocation for security in reliability systems, European Journal of Operational Research, 181, (2007), 773-786], the problem of determining resource allocation in series-parallel systems (SPSs) is considered. The results for this problem are based on the results for the least-expected cost failure-state diagnosis problem. In this note, it is demonstrated that the results for the least-expected cost failure-state diagnosis problem for SPSs in Azaiez and Bier [2007] are incorrect. In addition relevant results that were not cited in the paper are summarized.

Key words: Sequential testing, series-parallel systems.

1 Introduction

In Azaiez and Bier [2007], the problem of determining an optimal allocation of resources to the individual components of an SPS for security under a limited budget is considered. In order to develop a solution methodology for this problem, the authors use certain results for the least-expected cost failure-state diagnosis problem for SPSs. In this paper we will present instances of the least-expected cost failure-state diagnosis problem for which the algorithm proposed in Azaiez and Bier [2007] produce non-optimal solutions. In fact we will demonstrate that the algorithm proposed in Azaiez and Bier [2007] can produce arbitrarily bad solutions for certain instances. In addition, we will summarize the results from the literature that were not cited by Azaiez and Bier [2007] related to the least-expected cost diagnosis of SPSs.

Let us first recall the definition of an SPS. The most basic SPSs are the simple series (parallel) systems where the system functions when all (at least one) components function. More complicated SPSs can be obtained by a series (or parallel) connection of other SPSs. The least-cost diagnosis problem for an SPS requires a strategy to find whether the SPS is functioning or not by testing the individual components of the SPS, where c_i is the cost of testing component i and p_i is the probability that component i functions. It is assumed that the components fail or function independent of each other.

It is easy to show that it is optimal to test the components of a series (parallel) system in increasing order of $c_i/(1-p_i)$ (c_i/p_i) by an interchange argument. This result has been published in various papers, the earliest to our knowledge being Mitten [1960]. We shall say that the simple series (parallel) system has depth one. The depth of any other SPS is 1+max{the depth of any maximal subsystem}. For instance the SPS in figure 1 has depth 3 since it consists of two maximal subsystems.These two maximal subsystems are connected in series with each other. One of them consists of component 1, with depth 1 and the other consists of components 2, 3 and 4 with depth 2. The latter maximal subsystem itself consists of two maximal subsystems of depth 1. One of them is the series system consisting of component 2 and the other is the series system consisting of components 3 and 4.

The following lemma and theorem are mentioned in section 4.1 of Azaiez and Bier [2007] and later used in section 5 for the main results of the paper. (In Azaiez and Bier [2007] these are referred to as Lemma 4.1 and Theorem 4.1.) Specifically, these results are used in section 5.3 for general combined series/parallel systems.

Lemma 1 *Consider any ordered series or parallel subsystem $S = (S_1, ..., S_n)$. Then in order to minimize the expected testing cost, testing of any basic constituent S_i must be performed to completion before moving on to testing of another basic constituent with a subscript higher than i.*

Theorem 1 *Consider a combined series/parallel system S, ordered according to the initialization algorithm. Then, the optimal testing policy that minimizes the expected testing cost is to follow the orderings specified in the initialization algorithm. Moreover, if a basic constituent S_j^i of subsystem $S_j = (S_j^1, ..., S_j^n)$ is to be tested, then it should be tested to completion before moving on to testing of basic constituent S_j^{i+1} of that subsystem (or testing of some other subsystem), if needed. In this case, the optimal expected testing cost of the system will equal C(S), as computed in the above algorithm.*

In Azaiez and Bier [2007], Lemma 1 is used to prove Theorem 1. The counter example presented below shows that Lemma 1 and Theorem 1 are incorrect as stated. Obviously, for some SPSs they may hold. In particular, for SPSs with depth 1 or 2, they are correct (Ben-Dov [1981], Boros and Ünlüyurt [2000]). In addition, the strategy produced by *Initialization* could be a good heuristic algorithm for more general SPSs. As mentioned before, these results are for general combined series/parallel systems. In the paper, there are also results for simple series and simple parallel systems that are correct.

The *Initialization* algorithm mentioned above essentially does the following. Starting at the deepest level of the SPS where we have simple series (or parallel) systems, these simple series (parallel) systems are replaced with a single component whose testing cost is the optimal testing cost of the simple series (parallel) system and whose probability of functioning (failing) is the product of the functioning (failing) probabilities of the individual components. This operation is continued until a simple series (parallel) system is obtained. The theorem essentially states that it is optimal to test whole system by testing the maximal subsystems one by one to completion in the optimal order for the final simple series (parallel) system.

For instance, for the SPS in figure 1, let us assume that $\mathbf{c} = (5.0, 1.8, 1.0, 6.0)$ and $\mathbf{p} = (0.5, 0.5, 0.5, 0.5)$ where c_i is the cost of testing component i and p_i is the probability that component i functions. *Initialization* algorithm first replaces the simple series system consisting of components 3 and 4 by a single component say component 34 whose testing cost is $1.0 + 0.5 \times 6 = 4.0$, and whose probability of functioning is $0.5 \times 0.5 = 0.25$. Then it replaces the simple parallel system consisting of components 2 and 34, by a component say 234, with testing cost $1.8 + 0.5 \times 4.0 = 3.8$ and whose probability of failing is $(1 - 0.5) \times (1 - 0.25) = 0.375$. The resulting SPS is a simple series system that consists of components 1 and 234. The $c_i/(1 - p_i)$ ratio is $5/0.5 = 10$ for component 1 and the ratio is $3.8/0.375 = 10.13$ for component 234. So the total cost of inspecting these in the optimal order gives $5 + 0.5 \times 3.8 = 6.9$ and the testing sequence is $(1, 2, 3, 4)$.

In the next section, we will provide a better strategy than this one for this particular example. In this better strategy, the maximal subsystems are not tested to their entirety. Hence Lemma 1 and Theorem 1 are incorrect as stated. In fact, we will provide a series of examples such that applying the *Initialization* algorithm misses the optimal solution by a factor k for any given k where the number of individual components is a polynomial function of k. Finally in section 3 we will conclude by a brief discussion.

2 Counter Examples

2.1 Counter example 1

Consider the SPS shown in figure 1 with $\mathbf{c} = (5.0, 1.8, 1.0, 6.0)$ and $\mathbf{p} = (0.5, 0.5, 0.5, 0.5)$ where c_i is the cost of testing component i and p_i is the probability that component i functions. If the *Initialization* algorithm of Azaiez and Bier [2007] is applied to this problem, the claimed permutation turns our to be (1,2,3,4) and the expected cost of this strategy turns out to be 6.9 as computed in the previous section. Let us consider the following strategy that starts by testing component 2 first. If component 2 functions, then component 1 is tested. If component 2 fails, then the resulting system is a simple series system and the optimal ordering can be obtained by checking the $c_i/(1-p_i)$ ratios as (3,1,4). The expected cost of this strategy is $c_2 + p_2 c_1 + (1-p_2)(c_3 + p_3 c_1 + p_3 p_1 c_4) = 6.8$. Actually, it can be shown that this strategy is the only optimal strategy.

So the strategy produced by *Initialization* is not optimal. This example is presented in Boros and Ünlüyurt [2000].

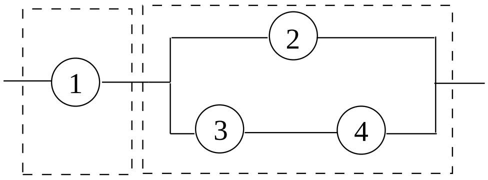

Fig. 1. SPS for the counter example

Let us say that the testing strategies that inspect relevant components in a fixed order are called permutation strategies. A relevant component is one whose functionality affects the functionality of the whole system, given the states of the already inspected components. It is clear that the strategies produced by *Initialization* are permutation strategies. In general, strategies can naturally be described by binary trees where the node corresponds to the component that will be tested and the two branches correspond to the failing and functioning states of the component. In this respect, permutation strategies are very easy to represent. One just needs to keep a permutation of the components rather than a binary tree. So it may be interesting to find the best permutation strategy. The strategy produced by *Initialization* for the counter example was (1,2,3,4). When we examine the better strategy, we observe that it is also a permutation strategy corresponding to (2,1,3,4). So *Initialization* does not always produce the best permutation strategy either.

Let us note that the better strategy starts by testing the maximal subsystem consisting of components $(2, 3, 4)$. If component 2 fails, then the strategy switches to the maximal subsystem consisting of only component 1. So it does not test a maximal subsystem to its entirety as stated in Lemma 1 and Theorem 1.

In fact, in Boros and Ünlüyurt [2000], examples demonstrating the following

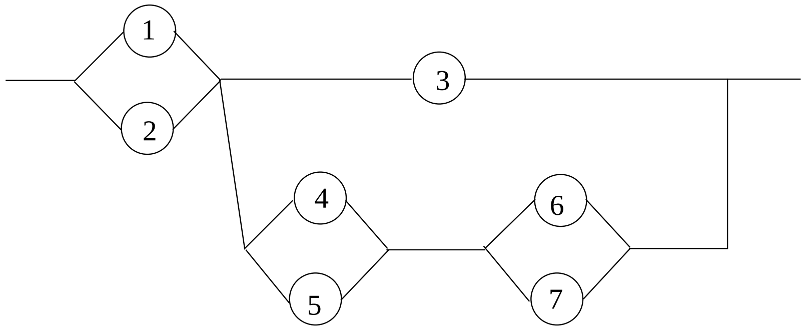

Fig. 2. SPS for the second counter example

are also provided.

a) Even when all components are identical in terms of testing costs and all probabilities are 0.5, one can construct an example SPS for which the *Initialization* algorithm produces a non-optimal testing strategy.
b) One can construct an instance of an SPS where no permutation strategy is optimal.

2.2 Counter example 2

In this section, we show that it is possible to construct an SPS for which the algorithm proposed in Azaiez and Bier [2007] misses the optimal solution value by a factor of k for any given k, Ünlüyurt [1999].

Let us consider the SPS in figure 2. We can make the following observations regarding this SPS.

a) There exists a p such that when $p_i = p$ for all components, the probability that the whole system functions is also p and $0 < p < 1$. (One possible such value of p is $\frac{3-\sqrt{5}}{2}$, found using MAPLE. This value is one solution of the equation $[1-(1-p)^2]\{1-(1-p)[1-(1-(1-p)^2)^2]\}$.)

b) There exists an ϵ such that when the costs are defines as $c_1 = 5$, $c_2 = \epsilon$, $c_3 = 1$, $c_4 = \epsilon$, $c_5 = 1$, $c_6 = \epsilon$ and $c_7 = 6$ and the probabilities are defined as in (a), *Initialization* does not produce an optimal testing strategy. One can achieve this by choosing an ϵ such that the second subsystem is inspected first. For instance $\epsilon = 0.1$ satisfies this condition. In that case component 3 has to be inspected first. If it is functioning, one ends up with a non-optimal strategy for the residual system. It is easy to see that an optimal strategy should switch to the first subsystem that consists of components 1 and 2.

c) In this SPS, every individual component belongs to a parallel subsystem.

Let C be the expected cost of the strategy induced by the algorithm *Initialization* and let C_{better} be the cost of the better strategy presented in property (b) above where $C_{better} = (1-\sigma)C$ for some $0 < \sigma < 1$.

Let us now consider another SPS that is obtained by replacing each individual component of the original SPS by a copy of the original SPS, where the costs of individual components are multiplied by c_i/C where c_i is the testing cost of the component that is being replaced and C is the cost of the strategy induced by the algorithm *Initialization*.

First of all, in this new SPS, each copy of the original SPS is a maximal subsystem due to property (c) and since the original system is globally series. This means that the ratios will be calculated for each copy of the original SPS. Let us now consider one such copy, say the one that replaces component i. Due to property (a), the probability that this subsystem works is p. Moreover, the expected cost of the strategy induced by *Initialization* will be c_i because of the fact that all costs are multiplied by c_i/C. In turn, this means that if

Initialization is applied to this SPS, the total expected cost will still be C.

On the other hand, if we apply the better algorithm to each copy of the original SPS and replace the subsystems by a single component whose expected cost and probabilities are calculated correspondingly, we obtain the original SPS, where each component has cost $(1-\sigma)c_i$. If we apply the same strategy again the total expected cost would be $(1-\sigma)^2 C$.

If we apply this replacing operation iteratively α times, then the strategy induced by *Initialization* will have cost C and the other strategy (that tests each copy by the better algorithm) will have cost $(1-\sigma)^\alpha C$. In order to create an instance for which the strategy induced by *Initialization* misses the optimal solution by a factor of k, one needs to iterate this replacing operation for $\alpha = \frac{-logk}{log1-\sigma}$ times. The resulting SPS will have polynomially many components in k.

3 Conclusion

In the previous section we showed that, Lemma 4.1 and Theorem 4.1 of Azaiez and Bier [2007] are incorrect. In fact, we provided an instance where it can be arbitrarily bad. On the other hand, it is shown in Boros and Ünlüyurt [2000] that the testing strategy induced by *Initialization* is optimal for 2-level deep general SPSs and 3-level deep SPSs that consists of identical components ($c_i = c$ and $p_i = p$). This is no longer true if we have 4-level deep SPSs consisting of identical components. As a matter of fact, one could improve the *Initialization* algorithm by recomputing all ratios after determining the next component to test. Even this improvement does not make the algorithm optimal as can be seen in case of the SPS in figure 1. The least-expected cost diagnosis problem has been studied for systems other than SPSs. A review of the results and applications can be found in Ünlüyurt [2004].

References

M.N. Azaiez and V.M. Bier. Optimal resource allocation for security in reliability systems. *European Journal of Operational Research*, 181:773–786, 2007.

Y. Ben-Dov. Optimal testing procedures for special structures of coherent systems. *Management Science*, 27(12):1410–1420, 1981.

E. Boros and T. Ünlüyurt. Sequential testing of series-parallel systems of small depth. In M. Laguna and J.L.G. Velarde, editors, *Computing Tools for Modeling, Optimization and Simulation*, pages 39–74. Kluwer, 2000.

L.G. Mitten. An analytic solution to the least cost testing sequence problem. *The Journal of Industrial Engineering*, page 17, January-February 1960.

T. Ünlüyurt. Sequential testing of complex systems: A review. *Discrete Applied Mathematics*, 142,(1-3):189–205, 2004.

T. Ünlüyurt. *Sequential testing of complex systems.* PhD thesis, Rutgers University, 1999.