DIMACS Technical Report 99-24
April 1999

Sequential Testing of Series-Parallel Systems of Small Depth

by

Endre Boros[1,2]
RUTCOR, Rutgers University,
640 Bartholomew Rd.
Piscataway, NJ, 08854-8003

Tonguc Unluyurt[3,4]
RUTCOR, Rutgers University,
640 Bartholomew Rd.
Piscataway, NJ, 08854-8003

[1]Permanent Member
[2]This research was partially supported by the ONR (Grant N00014-92-J-1375) and by DARPA (Contract Number N66001-97-C-8537)
[3]Affiliated Graduate Student Member
[4]This research was partially supported by the ONR (Grant N00014-92-J-1375) and by DARPA (Contract Number N66001-97-C-8537)

ABSTRACT

We consider the problem of testing sequentially the components of a multi component system, when the testing of each component is costly. We consider a polynomial time testing policy for *series-parallel systems*, proposed in the literature. Generalizing published results, we prove that the considered policy is cost-minimal in the average case sense, for two sub-families of series-parallel systems. We also demonstrate via examples that neither this algorithm nor some of its improved versions are optimal for general series-parallel systems, disproving some published claims.

1 Introduction

A *multi-component system* is composed of a set $N = \{u_1, u_2, \cdots, u_n\}$ of n components, each of which can be either *working* or *failing*. A *state vector* $\mathbf{x} = \{x_1, x_2, \cdots x_n\}$ is a binary vector, recording the state of the system, i.e. $x_i = 1$ if component u_i is working and $x_i = 0$ otherwise, for $i = 1, \cdots, n$. For the multi-component system itself, we shall distinguish two states: *working* states are those in which the system functions correctly, and all other states are called *failing*. The *structure function* of the system is a binary mapping f characterizing the working states, i.e. $f(\mathbf{x}) = 1$ when the system is functioning correctly, and $f(\mathbf{x}) = 0$ otherwise. Let us denote such a system by $\Gamma(N, f)$. We shall consider *monotone* (sometimes called *coherent*) systems, i.e. for which $f(\mathbf{x}) \geq f(\mathbf{y})$ holds whenever $\mathbf{x} \geq \mathbf{y}$ (componentwise). In other words, by fixing some of the failing components of a working monotone system, one cannot make it fail.

As a typical example for such multi-component systems we shall consider telecommunications networks between two specified nodes, called the terminals. In these systems the components are the links forming the network, and the system is working if one can establish a working connection between the terminals. For example, the network $\Gamma(N, f)$ in figure 1(a) consists of four components $N = \{u_1, u_2, u_3, u_4\}$, and it is working if u_1 is working, or if u_2 and at least one of u_3 and u_4 are working. As customary, we shall represent the structure function by a logical expression in terms of the states of its components. For this example we have

$$f(x_1, x_2, x_3, x_4) = x_1 \vee (x_2 \wedge (x_3 \vee x_4)) . \tag{1}$$

In this paper, we shall consider *series-parallel systems* (or SPSs, in short). An SPS is a specially structured network between two terminal nodes, called the *source* and the *sink*. The simplest SPS consists of only one component: a single link connecting the source to the sink. All other SPSs can be defined recursively as a series or parallel connection of smaller SPSs. A *series connection* identifies the sink of one SPS with the source of the another one, while a *parallel connection* identifies the sources and the sinks of the two systems. Formally, if $\Gamma(N_1, f_1)$ and $\Gamma(N_2, f_2)$ are two systems with no common components, then $\Gamma(N_1 \cup N_2, f_1 \vee f_2)$ is their parallel connection, and $\Gamma(N_1 \cup N_2, f_1 \wedge f_2)$ is their series connection. All smaller SPSs, appearing recursively in the above definition, are called the *subsystems* of the considered SPS.

For instance, in figure 1, the second network is not an SPS, since it cannot be obtained by the recursive procedure described above, while the first one is an SPS, formed by a parallel connection of the two subsystems $\{u_1\}$ and $\{u_2, u_3, u_4\}$. The latter one itself is a series connection of two smaller subsystems, $\{u_2\}$ and $\{u_3, u_4\}$, where this last subsystem is formed again by a parallel connection of the two single component subsystems $\{u_3\}$ and $\{u_4\}$. Let us observe that the expression in (1) corresponds to the recursive definition of this system, and in particular, the brackets enclose its (non-singleton) subsystems.

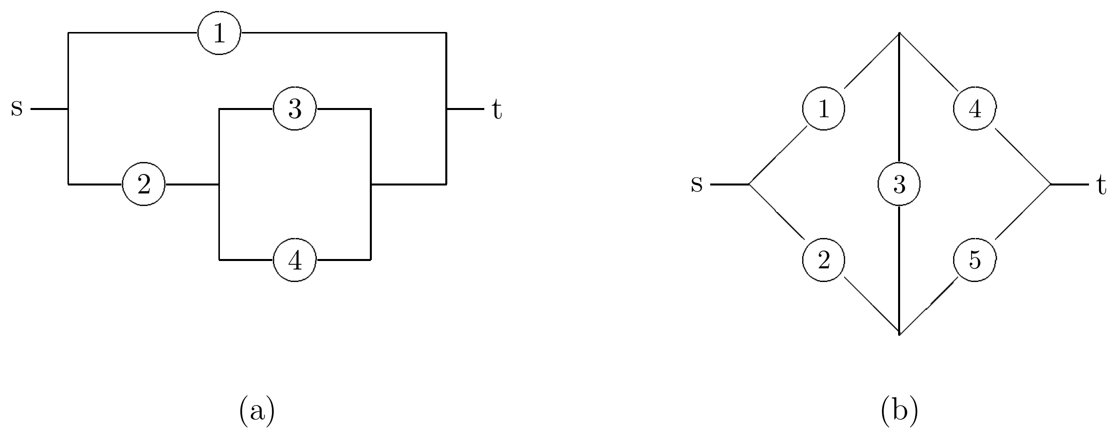

Figure 1: Examples for an SPS (a), and a non series-parallel network (b).

Given a multi-component system $\Gamma(N, f)$, let us consider the task of testing the functionality of Γ by inspecting its components one-by-one. It is quite evident that one may complete such a task without inspecting all the components of the given system. For example, after inspecting the first two components of the system of figure 1(a), and finding that $x_1 = x_2 = 0$, one can stop, and conclude that the system fails, regardless of the states of the remaining two components. Such a possibility can be very important in cases when the inspection is a costly procedure (i.e. it costs money, takes time, etc.) It is apparent that inspecting the components in different orders might result in different inspection costs.

Given a system $\Gamma(N, f)$ let us denote by $\Gamma|_{x_i=\alpha}$ the (restricted) system obtained from $\Gamma(N, f)$ by fixing component u_i at state $\alpha \in \{0, 1\}$, or in other words, $\Gamma|_{x_i=\alpha}$ represents the system after we learn that component $u_i \in N$ is in state α. Clearly, such a restricted system has at least one less "uncertain" components to depend on, and its system function can be obtained from f by substituting the value α for x_i. For example, if $\Gamma(N, f)$ is the system of figure 1(a), i.e. $N = \{u_1, u_2, u_3, u_4\}$ and f is given by (1), then $\Gamma|_{x_1=1} = \Gamma(\{u_2, u_3, u_4\}, 1)$, i.e. a system which works regardless of the states of its remaining components; and $\Gamma|_{x_3=0} = \Gamma(\{u_1, u_2, u_4\}, x_1 \vee (x_2 \wedge x_4))$, etc.

Since inspection is assumed to be performed sequentially, let us consider first the problem of selecting the next component for inspection. Let us define an inspection *strategy* $\mathcal{S}$ (or simply a strategy) as a rule which either recognizes if the system's state does not depend on its components or, if it does, selects a next component to be inspected. Such a strategy then can be applied, recursively, until it returns the state of the system.

For example, the rule $\mathcal{S} =$*choose the component with the smallest index, as long as the system depends on its components* is a possible (inspection) strategy. Let us e.g. apply this rule to the system of figure 1(a), and let us assume that the (hidden) current state of this system is $\mathbf{x} = (1, 1, 0, 1)$. Strategy $\mathcal{S}$ will, at first, select u_1 for inspection, and after testing this component we obtain the restricted system of $\Gamma(\{u_2, u_3, u_4\}, 1)$. Since this system does not depend on its components, $\mathcal{S}$ will return that the system is working, and the inspection terminates after one step. Let us remark that for a different hidden state, the same strategy may end up inspecting more components.

Let us denote by c_i the cost of inspecting component $u_i \in N$ of a system $\Gamma(N, f)$, and let $\mathbf{c} = (c_1, ..., c_n)$. Let us define the *cost of a strategy* $\mathcal{S}$ if applied to system $\Gamma(N, f)$ in the current state of $\mathbf{x}$ by $C(\mathcal{S}, \mathbf{x}) = \sum_{i \in I(\mathbf{x})} c_i$, where $I(\mathbf{x}) \subseteq \{1, 2, ..., n\}$ denotes the indices of the components inspected by $\mathcal{S}$ in this situation. To compare different strategies, one customary possibility is to consider the unknown state $\mathbf{x}$ of the system as a random variable, and use the expected value $EC(\mathcal{S}) = \mathrm{Exp}[C(\mathcal{S}, \mathbf{x})]$ as a measure of quality for a strategy.

More precisely, given a multi-component system $\Gamma(N, f)$, we shall assume that a cost vector $\mathbf{c} = (c_1, \cdots, c_n)$, where c_i denotes the cost of inspecting component $u_i \in N$, and a probability vector $\mathbf{p} = (p_1, \cdots, p_n)$, where p_i denotes the a priori probability that component u_i is working ($x_i = 1$), are all known, and that the components work or fail independently.

Let us compute the expected cost of the strategy mentioned above, i.e. $\mathcal{S} =$*choose the component with the smallest index, as long as the system depends on its components*, for the network in 1(a), when $\mathbf{p} = (0.5, 0.5, 0.5, 0.5)$ and $\mathbf{c} = (1, 2, 3, 4)$. For binary vectors of the type $\mathbf{x} = (1, *, *, *)$, where $*$ means the value can be 0 or 1, the strategy inspects only u_1. The probability that we have such a vector is $p_1 = 0.5$ and $C(\mathcal{S}, \mathbf{x}) = c_1 = 1$. Similarly, for $\mathbf{x} = (0, 0, *, *)$, $C(\mathcal{S}, \mathbf{x}) = c_1 + c_2 = 3$ and the probability is $q_1 q_2 = 0.25$, For $\mathbf{x} = (0, 1, 1, *)$, $C(\mathcal{S}, \mathbf{x}) = c_1 + c_2 + c_3 = 6$ and the probability is $q_1 p_2 p_3 = 0.125$ and finally for $\mathbf{x} = (0, 1, 0, *)$ $C(\mathcal{S}, \mathbf{x}) = c_1 + c_2 + c_3 + c_4 = 10$ and the probability is $q_1 p_2 q_3 = 0.125$. Hence for the total expected cost $EC(\mathcal{S}) = \mathrm{Exp}[C(\mathcal{S}, \mathbf{x})]$ we get $EC_(\mathcal{S}) = 0.5 + 3 \times 0.25 + 6 \times 0.125 + 10 \times 0.125 = 3.25$.

We shall consider the problem of finding a strategy *minimizing* $EC(\mathcal{S})$, and in particular, we shall consider this problem for special types of SPSs. The objective of minimizing the expected cost of inspection corresponds to the practical desire of minimizing the total cost of inspection over a long period of time, during which the same strategy is applied to the same system many times.

Let us remark that an inspection strategy, in principle, could always involve the testing of all components, simply by continuing testing even if the functionality of the system has already been determined. Let us call a strategy *reasonable* if it stops as soon as the restricted system does not depend on its components.

2 Series-Parallel Systems

The *depth* of an SPS is a measure of the number of series and/or parallel combinations in the system and is defined as follows: The depth of a simple series or a simple parallel system is 1. The depth of any other SPS is $1 + \max\{$depth of proper subsystems$\}$. Thus, every SPS is either a parallel or series connection of other SPS whose depths are at least 1 less than the original SPS. For example, the SPS in figure 2 has depth 4.

It will be very useful to represent an SPS $\Gamma(N, f)$ by a rooted tree structure $T\Gamma(N, f)$

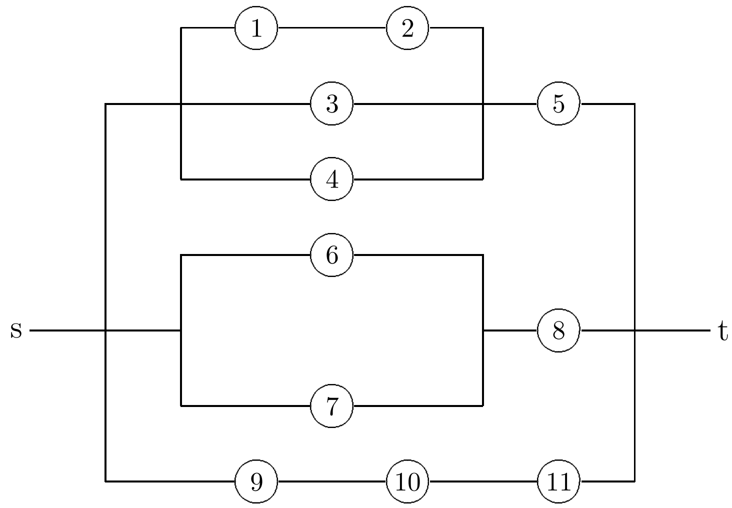

Figure 2: An SPS of depth 4

in the following manner: Each node of the tree represents a subsystem. In particular, the root represents the system itself, while the leaves of the tree correspond to the components. Each node has as many children as its maximal subsystems, each of which corresponds to one of those subsystems. Since in an SPS subsystems are nested, i.e. two subsystems are either disjoint, or one contains the other, no component belongs to two different subsystem of the same level, hence the above definition indeed yields a rooted tree. Since an SPS is either a series or a parallel connection of its maximal proper subsystems, each node in this tree representation can be labeled by 'S' or 'P', corresponding to the type of the subsystem represented by that node. As we noted earlier, the system function of a subsystem corresponding to an 'S' node can be obtained by the conjunction of the system functions of its maximal subsystems, while a disjunction of those yields the system function of a 'P' node. For this reason, such trees are also called *AND-OR trees* (see e.g. [9].) For example the AND-OR tree corresponding to the SPS in figure 2 is given in figure 3, and its system function is

$$\begin{aligned} f(x_1, ..., x_{11}) = \quad & (((x_1 \wedge x_2) \vee x_3 \vee x_4) \wedge x_5) \\ & \vee ((x_6 \vee x_7) \wedge x_8) \\ & \vee (x_9 \wedge x_{10} \wedge x_{11}) \,. \end{aligned}$$

Note that the depth of the tree $T\Gamma(N, f)$ (i.e. the length of the longest path from the root to a leaf node) is equal to the depth of the SPS $\Gamma(N, f)$.

The Boolean *dual* f^d of a system function f is defined by $f^d(\mathbf{x}) = \overline{f}\,(\overline{\mathbf{x}})$, where $\overline{f} = 1 - f$ denotes the logical complementation, and $\overline{\mathbf{x}}$ denotes the binary vector obtained from $\mathbf{x}$ by complementing each of its components.

It is important to note that the Boolean dual f^d of the structure function of an SPS $\Gamma(N, f)$ is again a structure function of another SPS $\Gamma^d(N, f^d)$, which we will call the *dual system* for the original SPS. The dual system can be obtained simply by interchanging the

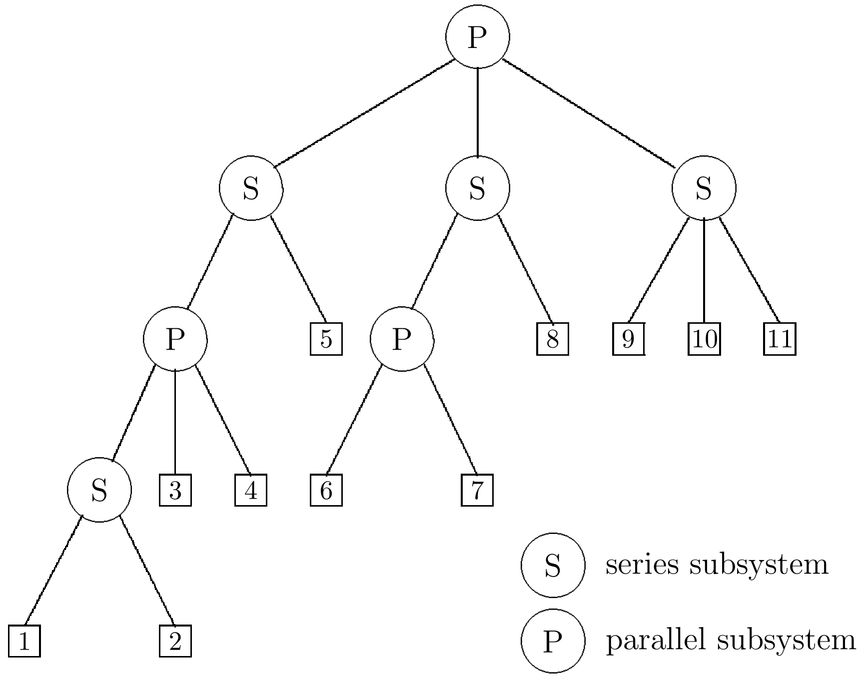

Figure 3: The tree representation of the SPS given in Figure 2.

$\vee$ (parallel) and $\wedge$ (series) signs in either the AND-OR tree representation, or in the system function (like in (1).) If the global problem is (series) parallel for the original SPS then the global problem for the dual is parallel (series). For example, the dual of the system given in figure 2 is shown in figure 4, and its system function is

$$\begin{aligned} f(x_1,...,x_{11}) = \quad & (((x_1 \vee x_2) \wedge x_3 \wedge x_4) \vee x_5) \\ & \wedge ((x_6 \wedge x_7) \vee x_8) \\ & \wedge (x_9 \vee x_{10} \vee x_{11}) \,. \end{aligned}$$

Let us note that, any SPS and its dual have the same depth. We also mention that all the results and algorithms in the following sections can straightforwardly be translated for the dual system, and hence without loss of generality, we can restrict our attention in the sequel to SPSs in which the global system is parallel.

It is important to note that in general, an optimal inspection strategy cannot be represented by a permutation of the components. It is frequently advantageous to inspect the remaining components in an order, which depends on the states of the already inspected components. The complete description of an optimal strategy can in fact be exponentially large in terms of n. Therefore, we shall use a more efficient representation of inspection strategies, namely by providing a selection algorithm, which for every system either recognizes the system's state, when it does not depend on its components, or will choose one

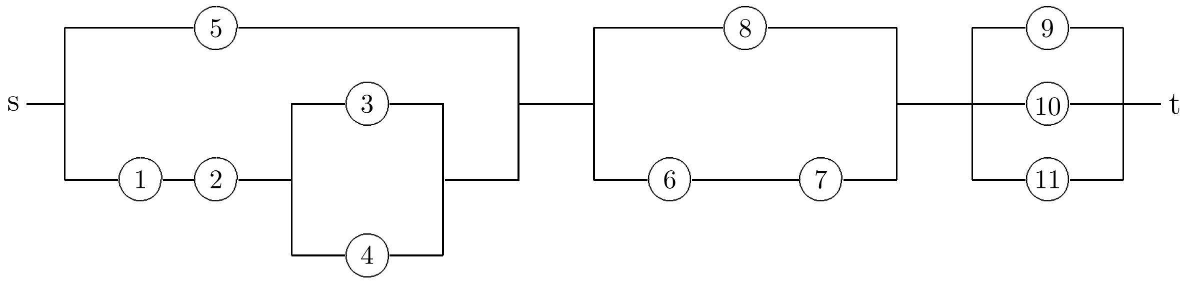

Figure 4: The dual system of the one shown in Figure 2.

of its components to be inspected next. Such a procedure then can be applied recursively, until the functionality of the system is determined. In this way, a selection algorithm defines implicitly an inspection strategy.

We shall say that such a selection algorithm is *optimal*, if it defines an optimal inspection strategy (i.e. one minimizing the expected cost of inspection.) We shall call such an inspection strategy *polynomial*, if the selection algorithm runs in polynomial time in the size of the system.

3 An Inspection Strategy for Series-Parallel Systems

A natural idea to find optimal strategies for an SPS, is to consider subsystems one by one, in some order, and inspect each of them optimally within themselves. If the global problem is a parallel one, then one can stop as soon as a working subsystem is found concluding that the SPS works. Indeed, this idea is the source for all the efficient strategies found in the literature. (see e.g. [9].)

Let us note that if we somehow order the maximal subsystems (i.e. those of depth 1) of a globally parallel system $\Gamma(N,f)$, and inspect them one by one independently in this order by a strategy $\mathcal{S}$ such that the expected cost of inspecting the i^{th} subsystem is EC_i, then the total expected cost of inspecting $\Gamma(N,f)$ by this strategy, $EC = EC(\mathcal{S})$ can be written as

$$EC = \sum_{i=1}^{m} Q_1 Q_2 ... Q_{i-1} EC_i, \tag{2}$$

where Q_i denotes the probability that the i^{th} subsystem is not working. The equality (2) holds, because $Q_1Q_2..Q_{i-1}$ is the probability that the i^{th} subsystem has to be inspected, implying that its additional expected cost of EC_i will be incurred.

For a simple parallel (series) system (i.e. a 1-level SPS), such a strategy can simply be described by a permutation of the components. Let σ and π be permutations of $\{1,2..,n\}$ such that

$$\frac{c_{\sigma_1}}{p_{\sigma_1}} \leq \frac{c_{\sigma_2}}{p_{\sigma_2}} \leq ... \leq \frac{c_{\sigma_n}}{p_{\sigma_n}} \quad \text{and} \quad \frac{c_{\pi_1}}{q_{\pi_1}} \leq \frac{c_{\pi_2}}{q_{\pi_2}} \leq ... \leq \frac{c_{\pi_n}}{q_{\pi_n}}, \tag{3}$$

where $q_i = 1 - p_i$ for $i = 1, ..., n$.

It is known, and can easily be shown that the optimal inspection strategy for a simple parallel system is to inspect its components in σ order until the state of the system is determined, while for a series system the order π is optimal (see e.g. [7, 5, 9, 1] and many others.) This (optimal) strategy has the following natural interpretation, let us say for a simple parallel system: Since we stop the inspection process as soon as we find a working component, it is very natural to begin inspection from low cost components whose probability of working is high, i.e. one for which the ratio c_i/p_i is the smallest.

Let us consider next globally parallel SPSs of depth 2. By the same token, one might want to inspect the subsystems in the increasing order of EC_i/P_i, where now EC_i denotes the expected cost of inspection, and P_i denotes the probability of working for the i^{th} subsystem. Ben-Dov [4] proposed such an algorithm, which essentially replaces each subsystem with a single component whose inspection cost is the optimal expected cost of that subsystem and whose probability of functioning is the probability that the subsystem functions. Then the resulting system becomes a simple parallel system, and the algorithm inspects the subsystems in the (optimal) order for depth 1 SPSs (i.e. in their σ order, in this case) without interrupting the inspection of a subsystem before its state is determined. Let us remark that, although this algorithm is optimal, the proof given in [4] is incomplete. We shall provide a correct proof of this fact in Section 4.

A very similar algorithm is proposed in [6] for general SPSs, and is claimed, mistakenly, to be optimal.

It is shown in [9] that among the inspection strategies which inspect subsystems one by one, recursively, without interruptions, the above procedure is indeed the best for globally parallel systems (and analogously, the one generalizing the simple series case is the best for globally series systems.)

We can however, demonstrate by a small example in Section 5 that an optimal strategy may require the interruption of the inspection of a subsystem, even for depth 3 SPSs.

An interesting special case is when all the components are identical, i.e. the cost of inspecting the components is the same c for all components, and all components function with the same probability p (and fails with probability $q = 1 - p$.) It is shown for this case in [3] that a generalization of the above mentioned algorithm for 2 level deep SPSs is optimal for some specially structured 3 level deep SPSs with identical components. We shall generalize these results in Section 5, and provide an optimal inspection strategy for all depth 3 SPSs with identical components.

Let us remark finally that in the special cases, considered in this paper, the optimal strategies can in fact be represented by a permutation of the components, and such a permutation can be computed in polynomial time.

Let us return to the idea of uninterrupted inspection of subsystems, which, as we mentioned above, is perhaps the most natural idea for generalizing the optimal algorithm for 2-level deep SPSs.

More generally, let $\Gamma(N, f)$ be an SPS of depth d, and let us consider its deepest nontrivial (i.e. level 1) subsystems, which are simple parallel or series systems. Let us replace each of these simple subsystems by a single component whose inspection cost is the optimal expected cost of the considered subsystem, and whose probability of working is the probability that the considered subsystem functions correctly. Since all these level 1 subsystems are simple series or parallel, these computations can be done efficiently by using the results cited above. In this way we obtain in one iteration a new SPS which has depth one less than the original system. Thus, by applying recursively this operation to the resulting SPSs, we arrive after d iterations to a simple system consisting of only one component.

In fact at each step in this procedure, the subsystems are "optimally" permuted, hence at the very end, all subsystems of $\Gamma(N, f)$, including its components, are ordered. Thus, this order induces a permutation $\rho \in \mathbb{S}_N$ of the components of $\Gamma(N, f)$, where $\mathbb{S}_N$ is the set of all permutations of N. We shall consider the inspection strategy, in which we inspect components in this ρ order, one by one, skipping those which become irrelevant.

Let us describe these computations at detail below.

SPS-Permutation$(f, \mathbf{p}, \mathbf{c})$

Input: A structure function f of a (globally parallel) SPS $\Gamma(N, f)$ specified by an AND-OR tree, the cost vector $\mathbf{c}$, and the probability vector $\mathbf{p}$.

Output: A permutation $\rho \in \mathbb{S}_N$ of the components.

Step 0: Consider the AND-OR tree of the SPS, and let k denote the depth of this tree. For all leaves a set $EC(a) = c_i$, $P(a) = p_i$, $Q(a) = 1 - P(a)$, and set the label $L(a) = (u_i)$ where u_i is the component corresponding to leaf a.

Step 1.1: For $j = k - 1$ to 1 and for each node a at level j, if node a is a series node then goto step 1.2, otherwise goto step 1.3.

Step 1.2: Let b_1, b_2, ..., b_t denote the children of a in the increasing order of their $\frac{EC(b)}{Q(b)}$ ratios. Let $EC(a) = \sum_{i=1}^{i=t} P(b_1) \cdots P(b_{i-1}) EC(b_i)$, and $P(a) = P(b_1)P(b_2) \cdots P(b_t)$. Goto step 1.4.

Step 1.3: Let b_1, b_2, ..., b_t denote the children of a in the increasing order of their $\frac{EC(b)}{P(b)}$ ratios. Let $EC(a) = \sum_{i=1}^{i=t} Q(b_1) \cdots Q(b_{i-1}) EC(b_i)$, and $P(a) = 1 - Q(b_1)Q(b_2) \cdots Q(b_t)$. Goto step 1.4.

Step 1.4 Set $Q(a) = 1 - P(a)$, and label node a with the concatenation of the labels of its children in the order calculated in the previous step, i.e. $L(a) = L(b_1), L(b_2), \cdots, L(b_t)$. Continue with Step 1.1.

Step 2: Output the label of the root $\rho = L(root)$.

Let us remark first that if the system function f is represented by a Boolean expression, or by an oracle, the corresponding AND-OR tree can efficiently be built (see e.g. [2, 8].

Let us also remark that the above procedure is essentially the same to the one considered in [4, 6, 9].

We shall consider two inspection strategies, based on the above procedure. The first one is, strategy $\mathcal{S}^\rho$, which *inspects the components one by one in their ρ order until the value of f is determined* (skipping those which do not effect f anymore, as a result of the states of some already inspected components.)

We shall also consider a variation, $\mathcal{S}^*$ of the above strategy, one that improves, if possible, on $\mathcal{S}^\rho$. Strategy $\mathcal{S}^*$ is induced by the selection procedure, in which we compute ρ, as before, and choose the first component $\rho_1 \in N$ to inspect next. The main difference between $\mathcal{S}^\rho$ and $\mathcal{S}^*$ is that in the latter, the permutation ρ is recomputed for the residual system, whenever a component is inspected.

As one can see easily, both of these strategies are reasonable, and can be carried out for any SPS in polynomial time. In particular, once the AND-OR tree is obtained the algorithm amounts to ordering the children of each node with respect to the appropriate ratios. If the number of leaves (which is the number of components) is n, this can be accomplished in $O(n \log n)$ time.

Our main results are as follows

Theorem 1 *Strategy $\mathcal{S}^\rho$ is optimal if applied to depth* 2 *SPSs.*

Theorem 2 *Strategy $\mathcal{S}^\rho$ is optimal if applied to depth* 3 *SPSs with identical components.*

We shall provide an example in Section 6 for a 4-level deep SPS with identical components, for which $\mathcal{S}^*$ is an optimal strategy, while $\mathcal{S}^\rho$ is not.

We shall also demonstrate in Section 6 by a small 3-level deep SPS that neither of these strategies are optimal, in general, even if applied only to 3-level deep SPSs. In fact the only optimal strategy for the provided example has to interrupt the inspection of a subsystem before completion, disproving hence the claim made in [6].

We shall also provide an example for a 3-level deep SPS, for which the optimal inspection policy cannot be described by a permutation of the components.

4 Series-Parallel Systems of Depth 2

In this section we show that the algorithm SPS-Permutation, proposed in section 3 is optimal for 2-level deep SPSs, as stated in Theorem 1.

This result is claimed by two other authors in [4] and [6], however, both proofs are incomplete. In [4], this result is derived from a lemma stating that "*All optimal procedures for testing a 2-level deep SPS finish determining the state of a subsystem before moving on to test another subsystem*", which, at this generality, is not true as shown by the example of Figure 5. For this 2-level deep SPS of four components, both permutations $\rho_1 = (1, 2, 3, 4)$ and $\rho_2 = (1, 3, 2, 4)$ induce optimal inspection policies, but the second one violates the above statement.

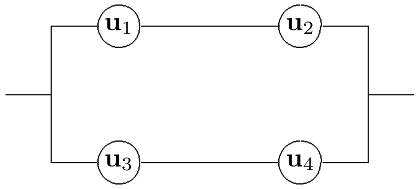

Figure 5: An SPS of four components, with $\mathbf{c} = (0, 1, 0, 1)$ and $\mathbf{p} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, for which both permutations $\rho_1 = (1, 2, 3, 4)$ and $\rho_2 = (1, 3, 2, 4)$ induce optimal inspection policies with the same $\frac{7}{8}$ expected cost.

Even if we assume that "all" is changed to an existential quantor in the above statement, the proof provided in [4] uses very general arguments, without detailed explanations, not utilizing the special structure of a 2-level deep SPS, and the example in Section 6 shows that already for 3-level deep SPSs, such a statement cannot be true.

It is important to point out again that, in principle, an optimal inspection strategy is not necessarily induced by a permutation, and it might continue testing with different orders of the remaining components, depending on the states of the inspected components (c.f. Section 6.) Both [4] and [6] seem to overlook this possibility. In [6] it is shown that an equivalent algorithm is optimal for 2-level SPSs with up to $n = 4$ components, and generalizes this by saying that "*if a subsystem is not inspected to completion (i.e. one switches to another subsystem before determining the state of a subsystem), one could permute components to decrease the total expected cost.*" However, as we noted above, permuting the components may not be well-defined for an inspection policy. The author in [6] goes even further, claiming the same for for SPSs of depth more than 2, which clearly fails, as shown by the counter example in Section 6.

In this section we prove Theorem 1 by showing that SPS-Permutation produces an optimal strategy for 2-level deep SPSs. Since the proof is quite technical, we shall need some further notations to ease the presentation.

Let us consider a 2-level deep SPS $\Gamma(N, f)$, consisting of the parallel connection of m simple series subsystems. Let us denote the i^{th} subsystem by $\Gamma_i(N_i, f_i)$, where $N_i = \{u_{i_1}, u_{i_2}, \cdots, u_{i_{l(i)}}\}$, and $f_i = \bigwedge_{j=1}^{l(i)} x_{i_j}$. We have $\sum_{i=1}^{m} l(i) = n$, and we assume $l(i) \geq 1$ for all subsystems $i = 1, ..., m$. With this notation, we have $N = \{u_{i_j} | j = 1, ..., l(i),\ i = 1, ..., m\}$ and

$$f(x_1, ..., x_n) = \bigvee_{i=1}^{m} \bigwedge_{j=1}^{l(i)} x_{i_j}.$$

As before, let us denote by c_{i_j} the cost of inspection of u_{i_j}, and by p_{i_j} be the probability that u_{i_j} is functioning correctly, and set $q_{i_j} = 1 - p_{i_j}$ for all components u_{i_j}, $j = 1, .., l(i)$, $i = 1, ..., m$. Let us assume further that the components in the subsystems are labeled in such a way that we have

$$\frac{c_{i_1}}{q_{i_1}} \leq \frac{c_{i_2}}{q_{i_2}} \leq .... \leq \frac{c_{i_{l(i)}}}{q_{i_{l(i)}}}, \quad for \quad i = 1, .., m. \tag{4}$$

Let $C_i = EC(\Gamma_i)$ be the optimal expected cost of inspection, and P_i be the probability of functioning correctly for subsystem Γ_i, and define $Q_i = 1 - P_i$. Since the i^{th} subsystem is simply a series system of independent components we have the following equalities (see e.g. [5].)

$$\begin{aligned} &C_i = c_{i_1} + p_{i_1} c_{i_2} + p_{i_1} p_{i_2} c_{i_3} + \cdots + p_{i_1} p_{i_2} \cdots p_{i_{l_{i-1}}} c_{i_{l(i)}} \text{ and} \\ &P_i = p_{i_1} p_{i_2} \cdots p_{i_{l(i)}}, \text{ and hence} \\ &Q_i = 1 - p_{i_1} p_{i_2} \cdots p_{i_{l(i)}}. \end{aligned} \tag{5}$$

Let us also assume that the subsystems of $\Gamma(N, f)$ are labeled such that the following inequalities hold:

$$\frac{C_1}{P_1} \leq \frac{C_2}{P_2} \leq .... \leq \frac{C_m}{P_m}. \tag{6}$$

Proof of Theorem 1. Let us observe that with the definition above, SPS-Permutation will output the permutation $\rho = (u_{1_1}, ..., u_{1_{l(1)}}, u_{2_1}, ..., u_{2_{l(2)}}, ..., u_{m_{l(m)}})$, or in other words, the strategy $\mathcal{S}^\rho$ will inspect the subsystems in the above order Γ_1, Γ_2, ..., Γ_m, and within each subsystem the components in the optimal order of that simple series subsystem, by always skipping those components which became irrelevant.

By introducing

$$Q(j) = \prod_{i=1}^{j} Q_i \tag{7}$$

and defining $Q(0) = 1$ we can write

$$C(\mathcal{S}^\rho) = \sum_{h=1}^{m} Q(h-1)C_h. \tag{8}$$

Let us now proceed with the proof of Theorem 1 by induction on the number of components, n. For $n \leq 2$ we either have a series system or a parallel system, hence in either case the claim follows, e.g. by the result of [5].

Let us assume next that the theorem is true for all SPSs with at most $n-1$ components and let us consider $\Gamma(N, f)$ as above, a 2-level deep SPS of n components.

Let $\mathcal{S}_i^k$ denote the strategy that inspects u_{i_k}, the k^{th} component of the i^{th} subsystem first and the residual SPS (of at most $n-1$ components) is inspected optimally, which according to our inductive hypothesis follows the permutation produced by SPS-Permutation for the residual system.

We claim that

$$\mathcal{S}_1^1 \text{ is the same as } \mathcal{S}^\rho \tag{9}$$

and that

$$EC(\mathcal{S}^\rho) \leq EC(\mathcal{S}_i^k) \tag{10}$$

for all $k = 1, ..., l(i)$ and $i = 1, ..., m$. It then will follow from this claim and from our induction hypothesis that $\mathcal{S}^\rho$ is optimal for n component 2-level deep SPSs, hence completing the proof of the theorem.

To verify our claims (9) and (10), let us consider first the changes in the i^{th} subsystem, where strategy $\mathcal{S}_i^k$ starts by inspecting component u_{i_k}. Let us note that if $x_{i_k} = 0$, then all other components of the i^{th} subsystem become irrelevant, and will hence be skipped by $\mathcal{S}_i^k$. In this case SPS-Permutation for the residual system $\Gamma|_{x_{i_k}=0}$ will output the same order as ρ, except without the components of the i^{th} subsystem,

$$\Gamma_1, ..., \Gamma_{i-1}, \Gamma_{i+1}, ..., \Gamma_m \tag{11}$$

and hence strategy $\mathcal{S}_i^k$ will inspect the rest of the components in their ρ-order according to our inductive hypothesis.

Let us now consider the case, when component u_{i_k} is found working. Let us denote by $C_i^k = EC(\Gamma_i|_{x_{i_k}=1})$ the optimal expected cost of testing subsystem i excluding its k^{th} component (i.e the series subsystem $(u_{i_1}, .., u_{i_{k-1}}, u_{i_{k+1}}, .., u_{i_{l(i)}})$), and let P_i^k denote the probability that $\Gamma_i|_{x_{i_k}=1}$ is working correctly. Thus, we have

$$\begin{aligned} P_i^k &= \frac{P_i}{p_{i_k}} \\ C_i^k &= c_{i_1} + p_{i_1} c_{i_2} + ... + p_{i_1} ... p_{i_{k-2}} c_{i_{k-1}} + \frac{p_{i_1} ... p_{i_k}}{p_{i_k}} c_{i_{k+1}} .... + \frac{p_{i_1} ... p_{i_{l_{i-1}}}}{p_{i_k}} c_{i_{l(i)}}. \end{aligned} \tag{12}$$

It is then easy to see that

$$C_i - p_{i_k} C_i^k = q_{i_k}(c_{i_1} + p_{i_1} c_{i_2} + ... + p_{i_1} p_{i_2} ... p_{i_{k-2}} c_{i_{k-1}}) + p_{i_1} ... p_{i_{k-1}} c_{i_k}. \tag{13}$$

¿From the above we can easily derive the following.

Lemma 1 *For a series (sub)system the $\frac{C}{P}$ ratio decreases after inspecting a working component.*

Proof. Indeed, after inspecting u_{i_k} in the i^{th} subsystem, and finding $x_{i_k} = 1$, we can write, by (12) and (13) that

$$\begin{aligned} \frac{C_i}{P_i} - \frac{C_i^k}{P_i^k} &= \frac{C_i - p_{i_k} C_i^k}{P_i} \\ &= \frac{q_{i_k}(c_{i_1} + p_{i_1} c_{i_2} + ... + p_{i_1} p_{i_2} ... p_{i_{k-2}} c_{i_{k-1}}) + p_{i_1} ... p_{i_{k-1}} c_{i_k}}{P_i} \\ &\geq 0, \end{aligned}$$

from which $\frac{C_i^k}{P_i^k} \leq \frac{C_i}{P_i}$ follows, concluding the proof. □

It follows now from Lemma 1 that in this case we have

$$\frac{C_1}{P_1} \leq ... \leq \frac{C_j}{P_j} \leq \frac{C_i^k}{P_i^k} \leq \frac{C_{j+1}}{P_{j+1}} ... \leq \frac{C_i}{P_i} \leq ... \leq \frac{C_m}{P_m} \tag{14}$$

for some index $j < i$. Thus, strategy $\mathcal{S}_i^k$ will inspect the rest of the components again essentially in their ρ order, except the components of the i^{th} subsystem, which are inspected in their optimal order $u_{i_1}, ..., u_{i_{k-1}}, u_{i_{k+1}}, ..., u_{i_{l(i)}}$ inserted into permutation ρ between the components of the j^{th} and $(j+1)^{st}$ subsystems.

¿From the above we can derive our first claim, (9), since after inspecting u_{1_1}, strategy $\mathcal{S}_1^1$ will follow the ρ order for the rest, according to (11) and (14), i.e. exactly like $\mathcal{S}^\rho$.

To verify our second claim, (10), let us apply formulae (5) to all series subsystems of $\Gamma|_{x_{i_k}=1}$ and $\Gamma|_{x_{i_k}=0}$, and observe that, by using (7), (12) and (14) we can write $EC(\mathcal{S}_i^k)$ in the following way:

$$C(\mathcal{S}_i^k) = c_{i_k} + q_{i_k}\left(\sum_{h=1}^{i-1} Q(h-1)C_h + \sum_{h=i+1}^{m} \frac{Q(h-1)C_h}{Q_i}\right)$$

$$+p_{i_k}\left(\sum_{h=1}^{j} Q(h-1)C_h + Q(j)C_i^k + \sum_{h=j+1}^{i-1} Q(h-1)Q_i^k C_h + \sum_{h=i+1}^{m} \frac{Q(h-1)Q_i^k C_h}{Q_i}\right)$$

By substituting $Q_i^k = 1 - \dfrac{P_i}{p_{i_k}}$ and factoring out the C_h factors from each terms, we get

$$\begin{aligned} C(\mathcal{S}_i^k) = c_{i_k} + \sum_{h=1}^{j} Q(h-1)C_h + p_{i_k}Q(j)C_i^k \\ + \sum_{h=j+1}^{i-1} Q(h-1)Q_i C_h + \sum_{h=i+1}^{m} Q(h-1)C_h \end{aligned} \tag{15}$$

Therefore, for the difference $\Delta = C(\mathcal{S}^\rho) - C(\mathcal{S}_i^k)$ it follows by (8) and (15) that

$$\Delta = -c_{i_k} - p_{i_k}Q(j)C_i^k + Q(i-1)C_i + P_i \sum_{h=j+1}^{i-1} Q(h-1)C_h.$$

Since $\dfrac{C_h}{P_h} \leq \dfrac{C_i}{P_i}$ holds for $h = j+1, .., i-1$, we obtain from the above that

$$\begin{aligned} \Delta &\leq -c_{i_k} - p_{i_k}Q(j)C_i^k + Q(i-1)C_i + C_i \sum_{h=j+1}^{i-1} P_h Q(h-1) \\ &= -c_{i_k} - p_{i_k}Q(j)C_i^k + C_i Q(j)\left(P_{j+1} + Q_{j+1}P_{j+2} + ... + Q_{j+1}..Q_{i-1}\right) \\ &= -c_{i_k} - p_{i_k}Q(j)C_i^k + C_i Q(j) \\ &= -c_{i_k} + Q(j)(C_i - p_{i_k}C_i^k). \end{aligned}$$

Applying (13) we further obtain

$$\begin{aligned}\Delta &\le -c_{i_k} + Q(j)(C_i - p_{i_k} C_i^k)\\ &\le -c_{i_k} + (C_i - p_{i_k} C_i^k)\\ &= -c_{i_k} + q_{i_k}(c_{i_1} + p_{i_1} c_{i_2} + \ldots + p_{i_1} \cdots p_{i_{k-2}} c_{i_{k-1}}) + p_{i_1} \cdots p_{i_{k-1}} c_{i_k}\\ &= (c_{i_1} + p_{i_1} c_{i_2} + \ldots + p_{i_1} \cdots p_{i_{k-1}} c_{i_k}) - (c_{i_k} + p_{i_k} c_{i_1} + \ldots + p_{i_k} p_{i_1} \cdots p_{i_{k-2}} c_{i_{k-1}}).\end{aligned}$$

The last expression is nothing else but the expected difference of inspecting the series system formed by the k components $u_{i_1},...,u_{i_k}$, first by the optimal strategy, according to (4), and second by another strategy which inspects the k^{th} component first. Hence $\Delta \le 0$ follows, proving our claim (10), and thus completing the proof of Theorem 1. □

5 3-level deep SPSs with Identical Components

Neither of the algorithms $\mathcal{S}^*$ and $\mathcal{S}^\rho$ proposed in section 3 are optimal for 3-level SPSs in general, as shown by the examples in Section 6. We shall show below that for some special cases they still produce optimal strategies.

In this section we consider 3-level deep SPSs consisting of identical components. Such systems arise in practical situations when the components of the system are similar with regard to inspection cost and probability of functioning. We can assume for such systems, without loss of generality that $c_i = 1$ and $p_i = p$ for all components u_i, $i = 1, ..., n$. We shall also assume that at the top level, the system $\Gamma(N, f)$ consists of m subsystems, each of which is formed by a series connection of $l(i)$ smaller parallel subsystems. We shall denote the i^{th} top level subsystem Γ_i by $(a_{i_1}, a_{i_2}, .., a_{i_{l(i)}})$ where a_{i_j} is the number of components in Γ_{i_j}, the j^{th} parallel subsystem of Γ_i. Thus, in particular, we have

$$\sum_{i=1}^{m} \sum_{j=1}^{l(i)} a_{i_j} = n.$$

Analogously, let us denote Γ also by the concatenation of these sequences, i.e. as

$$\Gamma = [(a_{1_1}, ..., a_{1_{m_1}}), (a_{2_1}, ..., a_{2_{m_2}}), \ldots, (a_{m_1}, ..., a_{m_{m_m}})].$$

For example, the system $[(2, 3), (1, 2, 2), (1, 1, 1, 3)]$ is shown in Figure (6).

Without loss of generality we can label the parallel subsystems at depth 2 such that

$$a_{i_k} \le a_{i_{k+1}}, \text{ holds for } i = 1, \ldots, m, \text{ and } k = 1, \ldots, l(i). \tag{16}$$

Note that for a sub-subsystem $\Gamma_{i_j} = (a_{i_j})$ i.e. a parallel connection of a_{i_j} identical components, we have for its probability of functioning and its optimal expected inspection cost

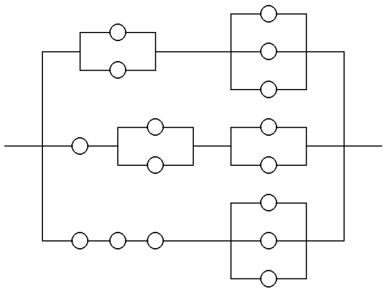

Figure 6: A 3 level deep series-parallel system $\Gamma = [(2,3),(1,2,2),(1,1,1,3)]$ consisting of 16 identical components.

that

$$P(\Gamma_{i_j}) = 1 - q^{a_{i_j}}$$
$$EC(\Gamma_{i_j}) = \frac{1 - q^{a_{i_j}}}{1 - q}. \tag{17}$$

Then, an optimal inspection strategy for the i^{th} subsystem can be determined, according to Theorem 1, by applying the SPS-Permutation algorithm, since each such subsystem is an SPS of depth at most 2. In other words, an optimal inspection strategy of such a subsystem inspects one by one the parallel sub-subsystems in increasing order of their $\dfrac{EC(\Gamma_{i_j})}{Q(\Gamma_{i_j})}$, which in this case is equivalent with the increasing order of the a_{i_j}s, until the correct state of the system is determined. Hence, by using (16) and (17), we get for the optimal expected cost of inspecting subsystem i that

$$C_i \stackrel{\text{def}}{=} EC(\Gamma_i) = \sum_{k=1}^{l(i)} \frac{1}{1-q} \prod_{l=1}^{k} (1 - q^{a_{i_l}}). \tag{18}$$

Let us denote by $P_i \stackrel{\text{def}}{=} P(\Gamma_i)$, the probability that the i^{th} subsystem is functioning properly, and let us introduce $Q_i = 1 - P_i$, for $i = 1, ..., m$. Hence, it follows by (17) that

$$P_i = \prod_{k=1}^{l(i)} (1 - q^{a_{i_k}}). \tag{19}$$

Let us assume finally, without loss of generality that the top level subsystems are labeled in such a way that

$$\frac{C_1}{P_1} \le \frac{C_2}{P_2} \le \cdots \le \frac{C_m}{P_m}. \tag{20}$$

We will show that the strategy $\mathcal{S}^\rho$, which inspects each subsystem optimally in this order is indeed optimal, as stated in Theorem 2.

For this proof, we shall need some more notations and a couple of useful technical lemmas. Let us first introduce the notation

$$F_i(j) = \frac{1}{(1 - q^{a_{i_j}})(1 - q^{a_{i_{j+1}}})...(1 - q^{a_{i_{l(i)}}})}, \tag{21}$$

which will substantially simplify the presentation of some of our computations in the sequel. Let us then observe that with this notation and by using (18) and (19) we can write

$$\frac{C_i}{P_i} = \frac{F_i(2) + F_i(3) + \cdots + F_i(l(i)) + 1}{1 - q}. \tag{22}$$

Let us next denote by $U^1(i_t)$ the residual subsystem when a component from the t^{th} parallel (sub)subsystem of the i^{th} subsystem is inspected and found to be functioning, i.e. $U^1(i_t) = (a_{i_1}, ..a_{i_{t-1}}, a_{i_{t+1}}, ..a_{i_{l(i)}})$, because the entire t^{th} parallel subsystem disappears when a component of it is found to be working. Similarly, let us denote by $U^0(i_t)$ the residual subsystem when a component from the t^{th} parallel subsystem with $a_{i_t} > 1$ is inspected and found to be failing, i.e. $U^0(i_t) = (a_{i_1}, .., a_{i_t} - 1, ..a_{i_{l(i)}})$, because in this case the size of the t^{th} parallel subsystem will be decreased by 1. To simplify notation, we shall drop sometimes the index t, if there can be no confusion.

Let $P_{i^\alpha} \stackrel{\text{def}}{=} P(U^\alpha(i_t))$ denote the probability that the system $U^\alpha(i_t)$ is functioning correctly, let $C_{i^\alpha} \stackrel{\text{def}}{=} EC(U^\alpha(i_t))$ denote the optimal expected inspection cost of the system $U^\alpha(i_t)$, and finally, let $Q_{i^\alpha} \stackrel{\text{def}}{=} 1 - P_{i^\alpha}$, for $\alpha \in \{0, 1\}$.

Finally, let us denote by $\tilde{C}_i$ the expected cost incurred when inspecting the i^{th} subsystem, by the following strategy: First we inspected a component from the t^{th} subsystem $\Gamma_{i_t} = (a_{i_t})$ and whatever is the functionality of this component, the optimal strategy (for 2 level deep SPSs) is applied to the remaining components of Γ_i.

Let us present next a series of technical lemmas, which we shall later use in our proof of Theorem 2.

Lemma 2 *We have the following relations:*

(a) $pQ_{i^1} + qQ_{i^0} = Q_i$;

(b) $pP_{i^1} + qP_{i^0} = P_i$;

(c) $\tilde{C}_i - C_i = 1 - \prod_{k=1}^{t-1}(1 - q^{a_{i_k}}) \geq 0$;

(d) $\frac{C_i}{P_i} - \frac{C_{i^1}}{P_{i^1}} = \frac{(q^{a_{i_t}}[F_i(2) + F_i(3) + \cdots + F_i(t)]) + F(t+1)}{1 - q} \geq 0$;

(e) $\frac{C_{i^0}}{P_{i^0}} - \frac{C_i}{P_i} = \frac{q^{a_{i_t}-1}[F_i(2) + F_i(3) + \cdots + F_i(t)]}{1 - q^{a_{i_t}-1}} \geq 0$, *whenever* $a_{i_t} > 1$.

Proof.

(a) The right hand side is the probability that the i^{th} subsystem Γ_i is not functioning. The left hand side is the same probability of the same event partitioned into two disjoint cases, corresponding to the working or failing state of one of the component in Γ_{i_t}.

(b) Analogous to [(a)].

(c) Let us introduce the notation $A_k = \frac{\prod_{l=1}^{k}(1-q^{a_{i_l}})}{1-q}$, for $k = 1, ..., l(i)$. Since the i^{th} subsystem is a simple 2-level deep series system, $\tilde{C}_i$ can be expressed, based on (16), (17), and the above notation, as

$$\tilde{C}_i = 1 + p\left(\sum_{k=1}^{t-1} A_k + \sum_{k=t+1}^{l(i)} \frac{1}{1-q^{a_{i_t}}} A_k\right) + q\left(\sum_{k=1}^{t-1} A_k + \sum_{k=t}^{l(i)} \frac{1-q^{a_{i_t}-1}}{1-q^{a_{i_t}}} A_k\right). \tag{23}$$

This is because, if the component inspected from $\Gamma_{i_t} = (a_{i_t})$ is functioning the parallel sub-subsystem Γ_{i_t} disappears, and if it is not functioning the number of components a_{i_t} of Γ_{i_t} decreases simply by 1. Observing that $p + q = 1$ and

$$p\frac{1}{1-q^{a_{i_t}}} + q\frac{1-q^{a_{i_t}-1}}{1-q^{a_{i_t}}} = 1$$

we can further obtain from (23) that

$$\tilde{C}_i = 1 + \sum_{k=1, k\neq t}^{l(i)} A_k + q\frac{1-q^{a_{i_t}-1}}{1-q^{a_{i_t}}} A_t. \tag{24}$$

Comparing this expression to the one in (18) for C_i, we get

$$\begin{aligned}\tilde{C}_i - C_i &= 1 + A_t\left(q\frac{1-q^{a_{i_t}-1}}{1-q^{a_{i_t}}} - 1\right)\\ &= 1 - A_t\frac{1-q}{1-q^{a_{i_t}}}\\ &= 1 - \prod_{l=1}^{t-1}(1-q^{a_{i_l}}) \geq 0.\end{aligned}$$

(d) Let us note that since $U^1(i_t) = (a_{i_1}, \ldots, a_{i_{t-1}}, a_{i_{t+1}}, \ldots, a_{i_{l(i)}})$ we have

$$\frac{C_{i^1}}{P_{i^1}} = \frac{[(1-q^{a_{i_t}})(F_i(2) + F_i(3) + \cdots + F_i(t))] + F_i(t+2) + \cdots + F_i(l(i)) + 1}{1-q}, \tag{25}$$

from which the desired equality follows by using (22).

(e) Similarly to (d), by noting that $U^0(i_t) = (a_{i_1}, \ldots, a_{i_t} - 1, \ldots, a_{i_{l(i)}})$, whenever $a_{i_t} > 1$, and hence we have

$$\frac{C_{i^0}}{P_{i^0}} = \frac{\frac{(1-q^{a_{i_t}})(F_i(2)+F_i(3)+\cdots+F_i(t))}{1-q^{a_{i_t}-1}} + F_i(t+1) + \cdots + F_i(l(i)) + 1}{1-q}, \tag{26}$$

from which the desired equality follows again easily by (22).

$\square$

Lemma 3 *Suppose $C_k \geq 0$ and $1 \geq P_k > 0$ are reals for $k = 1, 2$, such that $\dfrac{C_1}{P_1} \leq \dfrac{C_2}{P_2}$ holds. Then we have*

$$\frac{C_1}{P_1} \leq \frac{C_1 + Q_1 C_2}{1 - Q_1 Q_2} \leq \frac{C_2}{P_2}. \tag{27}$$

Proof. For the first inequality, let us start from the inequalities $1 - P_1 \geq 0$ and $C_1 P_2 - C_2 P_1 \leq 0$, which hold by our assumptions. Taking the product of these two and rearranging, we obtain

$$C_1 P_2 - C_1 P_1 P_2 \leq C_2 P_1 - C_2 P_1^2.$$

By adding $C_1 P_1$ to both sides we obtain

$$C_1(P_1 + P_2 - P_1 P_2) \leq P_1(C_1 + (1 - P_1)C_2)$$

implying the first inequality of (27), since $P_1 + P_2 - P_1 P_2 = 1 - Q_1 Q_2 > 0$.

Similarly, for the second inequality we can start with

$$C_1 P_2 \leq C_2 P_1$$

which follows form our assumptions. Then, by adding $C_2 P_2 Q_1$ to both sides we get

$$P_2(C_1 + Q_1 C_2) \leq C_2(P_1 + Q_1 P_2)$$

implying readily the second inequality of (27), since $P_1 + Q_1 P_2 = 1 - Q_1 Q_2 > 0$. $\square$

Lemma 4 *If $a_1 \leq a_2 \leq \cdots \leq a_t$ are nonnegative integers, $a_t \geq 1$, and $0 \leq q \leq 1$, then we have the inequality*

$$\frac{q^{a_t}}{1 - q^{a_t}} \sum_{k=1}^{t-1} \prod_{l=1}^{k} (1 - q^{a_l}) + \prod_{l=1}^{t} (1 - q^{a_l}) \leq 1. \tag{28}$$

Proof. First, let us note that since $q^{a_t} \leq q^{a_k}$ for $k = 1, 2, \ldots, t$ we have inequality:

$$q^{a_t}\left(\sum_{k=1}^{t-1}\prod_{l=1}^{k}(1-q^{a_l})+\prod_{l=1}^{t}(1-q^{a_l})\right) \leq (1-q^{a_1})\left(\sum_{k=1}^{t-1} q^{a_{k+1}}\prod_{l=2}^{k}(1-q^{a_l})+\prod_{l=2}^{t}(1-q^{a_l})\right).$$

Let us now consider $t-1$ independent events $E_2, \ldots, E_t$ such that the probability that event E_i happens is $P(E_i) = q^{a_i}$. Then the terms in the parenthesis on the right hand side are the probabilities of the pairwise disjoint events E_2, $\overline{E}_2 E_3$, $\ldots$, $\overline{E}_2 \cdots \overline{E}_{t-1} E_t$ and $\overline{E}_2 \cdots \overline{E}_{t-1} \overline{E}_t$, i.e. the events in which E_i is the smallest indexed event among these which happens, for $i = 2, ..., t-1$, and one in which none of these events happen. Since these cover all possibilities, their total probability is 1, hence the parentheses on the right hand side equals to 1. Thus, we obtain

$$q^{a_t}\left(\sum_{k=1}^{t-1}\prod_{l=1}^{k}(1-q^{a_l})+\prod_{l=1}^{t}(1-q^{a_l})\right) \leq (1-q^{a_1}) \leq (1-q^{a_t}).$$

completing the proof of the lemma. □

Proof of Theorem 2. We shall prove the theorem by induction on the number of components, n. If $n \leq 3$, then an SPS can be at most of depth 2 and hence Theorem 2 implies our claim. Let us assume that the proposed algorithm is optimal for all 3-level SPSs with at most $n-1$ components. We have to show that the expected cost incurred when we use the proposed strategy $\mathcal{S}^\rho$ is not more than that of any other strategy.

For this, let us consider first a "best" strategy, which begins by inspecting a component in the subsystem $\Gamma_{i_t} = (a_{i_t})$ for some i and t. Let us denote it by $\mathcal{S}^{i_t}$.

Without loss of generality we can assume that $a_{i_t} > a_{i_{t-1}}$ or $t = 1$. This is because if $a_{i_t} = a_{i_{t-1}}$ and $t > 1$, then we could simply relabel the subcomponents appropriately, since the components are identical.

For the ease of notation, we shall use again $Q(k) = \prod_{l=1}^{k} Q_l$, and $Q(0) = 1$. It is easy to see that the expected cost $EC(\mathcal{S}^\rho)$ of the strategy induced by the SPS-Permutation algorithm is:

$$EC(\mathcal{S}^\rho) = \sum_{k=1}^{m} Q(k-1) C_k \tag{29}$$

In strategy $\mathcal{S}^{i_t}$ we can assume, by the induction hypothesis that after the first element (from Γ_{i_t}) inspected, it proceeds by ordering the remaining subsystems in the increasing order of their $\frac{C}{P}$ ratios, and inspect them one by one optimally in that order until the functionality of the system is determined. In each of the subsystems, we have at most level 2 SPSs, and hence their optimal inspection can be carried out according to the SPS-Permutation algorithm, by Theorem 1. Let us now distinguish cases, according to the test result of the first inspection.

- If the first inspected component of Γ_{i_t} turns out to be functioning correctly, then by Lemma 2 (d) the order of inspection of the remaining subsystems is
$$\Gamma_1, \Gamma_2, \ldots, \Gamma_{j-1}, U^1(i_t), \Gamma_j, \ldots, \Gamma_{i-1}, \Gamma_{i+1}, \ldots, \Gamma_n$$
for some $j \leq i$, since the cost over probability ratio decreases for the i^{th} subsystem.
- If the first inspected component turns out to be failing, then by Lemma 2 (e) the order of inspection of the remaining subsystems is
$$\Gamma_1, \Gamma_2, \ldots, \Gamma_{i-1}, \Gamma_{i+1}, \ldots, \Gamma_{h-1}, U^0(i_t), \Gamma_h, \ldots, \Gamma_n$$
for some $h > i$, since the cost over probability ratio increases for the i^{th} subsystem.

Hence the expected cost of $\mathcal{S}^{i_t}$ can be written as:

$$\begin{aligned} EC(\mathcal{S}^{i_t}) = 1 + p \Bigg(& \sum_{k=1}^{j-1} Q(k-1)C_k + Q(j-1)C_{i^1} + \sum_{k=j}^{i-1} Q(k-1)Q_{i^1}C_k \\ & + \sum_{k=i+1}^{n} \frac{Q(k-1)Q_{i^1}}{Q_i} C_k \Bigg) \\ + q \Bigg(& \sum_{k=1}^{i-1} Q(k-1)C_k + \sum_{k=i+1}^{h-1} \frac{Q(k-1)}{Q_i} C_k + \frac{Q(h-1)}{Q_i} C_{i^0} \\ & + \sum_{k=h}^{n} \frac{Q(k-1)Q_{i^0}}{Q_i} C_k \Bigg) \end{aligned} \tag{30}$$

We shall show that

$$\Delta \stackrel{\text{def}}{=} EC(\mathcal{S}^\rho) - EC(\mathcal{S}^{i_t}) \leq 0. \tag{31}$$

Let us note that when we compute Δ, by (29) and (30), the terms including C_k for $k \leq j-1$ and $k \geq h$ cancel each other by Lemma 2 (a). Hence we can write Δ as

$$\begin{aligned} \Delta =& EC(\mathcal{S}^\rho) - EC(\mathcal{S}^{i_t}) \\ =& -1 + \left(\sum_{k=j}^{i-1} Q(k-1)C_k \right) [1 - pQ_{i^1} - q] \\ & + Q(i-1)C_i - pQ(j-1)C_{i^1} - q\frac{Q(h-1)}{Q_i} C_{i^0} \\ & + \left(\sum_{k=i+1}^{h-1} Q(k-1)C_k \right) \left[1 - p\frac{Q_{i^1}}{Q_i} - q\frac{1}{Q_i} \right]. \end{aligned} \tag{32}$$

Noting that $1 - pQ_{i^1} - q = pP_{i^1}$ and that $1 - p\frac{Q_{i1}}{Q_i} - q\frac{1}{Q_i} = -q\frac{P_{i0}}{Q_i}$ by Lemma 2 (a), we can further simplify the above, and obtain

$$\begin{aligned}\Delta = & -1 + pP_{i^1}\sum_{k=j}^{i-1} Q(k-1)C_k - q\frac{P_{i^0}}{Q_i}\sum_{k=i+1}^{h-1} Q(k-1)C_k \\ & + Q(i-1)C_i - pQ(j-1)C_{i^1} - q\frac{Q(h-1)}{Q_i}C_{i^0}.\end{aligned} \tag{33}$$

Let us now denote by $\Gamma_{a\to b}$ for some integers $a < b$ the SPS consisting of the subsystems $\Gamma_a, \Gamma_{a+1}, \dots, \Gamma_b$ connected in parallel. Hence the expected cost of inspecting this SPS by an SPS-Permutation based policy is

$$EC(\Gamma_{a\to b}) = \sum_{k=a}^{b} \frac{Q(k-1)}{Q(a-1)}C_k, \tag{34}$$

and probability that it works is

$$P(\Gamma_{a\to b}) = 1 - \frac{Q(b)}{Q(a-1)}. \tag{35}$$

With this notation we can see that

$$\begin{aligned}& pP_{i^1}\sum_{k=j}^{i-1} Q(k-1)C_k - pQ(j-1)C_{i^1} + pQ(i-1)C_{i^1} \\ & \qquad = pP_{i^1}Q(j-1)\left(1 - \frac{Q(i-1)}{Q(j-1)}\right)\left(\frac{EC(\Gamma_{j\to i-1})}{P(\Gamma_{j\to i-1})} - \frac{C_{i^1}}{P_{i^1}}\right),\end{aligned} \tag{36}$$

and that

$$\begin{aligned}& -q\frac{P_{i^0}}{Q_i}\sum_{k=i+1}^{h-1} Q(k-1)C_k - q\frac{Q(h-1)}{Q_i}C_{i^0} + qQ(i-1)C_{i^0} \\ & \qquad = qP_{i^0}Q(i-1)\left(1 - \frac{Q(h-1)}{Q(i)}\right)\left(\frac{C_{i^0}}{P_{i^0}} - \frac{EC(\Gamma_{i+1\to h-1})}{P(\Gamma_{i+1\to h-1})}\right).\end{aligned} \tag{37}$$

Using (36) and (37) we can rewrite (33) further, as

$$\begin{aligned}\Delta = \quad & -1 + pP_{i^1}Q(j-1)\left(1 - \frac{Q(i-1)}{Q(j-1)}\right)\left(\frac{EC(\Gamma_{j\to i-1})}{P(\Gamma_{j\to i-1})} - \frac{C_{i^1}}{P_{i^1}}\right) \\ & + qP_{i^0}Q(i-1)\left(1 - \frac{Q(h-1)}{Q(i)}\right)\left(\frac{C_{i^0}}{P_{i^0}} - \frac{EC(\Gamma_{i+1\to h-1})}{P(\Gamma_{i+1\to h-1})}\right) \\ & + Q(i-1)\left(C_i - pC_{i^1} - qC_{i^0}\right).\end{aligned} \tag{38}$$

Since we have

$$\frac{C_{i^1}}{P_{i^1}} \leq \frac{C_j}{P_j} \leq \cdots \leq \frac{C_i}{P_i} \leq \cdots \leq \frac{C_{h-1}}{P_{h-1}} \leq \frac{C_{i^0}}{P_{i^0}}$$

it follows by Lemma 3 that

$$\frac{EC(\Gamma_{j\to i-1})}{P(\Gamma_{j\to i-1})} \leq \frac{C_i}{P_i} \leq \frac{EC(\Gamma_{i+1\to h-1})}{P(\Gamma_{i+1\to h-1})}. \tag{39}$$

Thus, we get the following upper bound on Δ by (39) and (38):

$$\begin{aligned}\Delta \leq &-1 + \quad pP_{i^1}Q(j-1)\left(1-\frac{Q(i-1)}{Q(j-1)}\right)\left(\frac{C_i}{P_i}-\frac{C_{i^1}}{P_{i^1}}\right)\\ &+ qP_{i^0}Q(i-1)\left(1-\frac{Q(h-1)}{Q(i)}\right)\left(\frac{C_{i^0}}{P_{i^0}}-\frac{C_i}{P_i}\right)\\ &+ Q(i-1)\left(C_i - pC_{i^1} - qC_{i^0}\right).\end{aligned} \tag{40}$$

Noting that

$$P_{i^1} = \frac{(1-q^{a_{i_1}})}{F_i(2)(1-q^{a_{i_t}})}, \quad P_{i^0} = \frac{(1-q^{a_{i_1}})(1-q^{a_{i_t}-1})}{F_i(2)(1-q^{a_{i_t}})}, \tag{41}$$

and the equation $\dfrac{F_i(j)}{F_i(2)} = \displaystyle\prod_{k=2}^{j-1}(1-q^{a_{i_k}})$ we can obtain from Lemma 2 (d) and (e) that

$$\begin{aligned} pP_{i^1}(\frac{C_i}{P_i}-\frac{C_{i^1}}{P_{i^1}}) &= \frac{q^{a_{i_t}}}{(1-q^{a_{i_t}})}\left(\sum_{k=1}^{t-1}\prod_{l=1}^{k}(1-q^{a_{i_l}})\right)+\prod_{l=1}^{t-1}(1-q^{a_{i_l}})\\ qP_{i^0}\left(\frac{C_{i^0}}{P_{i^0}}-\frac{C_i}{P_i}\right) &= \frac{q^{a_{i_t}}}{(1-q^{a_{i_t}})}\sum_{k=1}^{t-1}\prod_{l=1}^{k}(1-q^{a_{i_l}})\end{aligned} \tag{42}$$

Finally, by using $\tilde{C}_i = 1 + pC_{i^1} + qC_{i^0}$ and Lemma 2 (c) we can write

$$C_i - pC_{i^1} - qC_{i^0} = C_i - \tilde{C}_i + 1 = \prod_{l=1}^{t-1}(1-q^{a_{i_l}}). \tag{43}$$

Hence, by introducing the notations

$$B = \frac{q^{a_{i_t}}}{(1-q^{a_{i_t}})}\sum_{k=1}^{t-1}\prod_{l=1}^{k}(1-q^{a_{i_l}}) \text{ and } S = \prod_{l=1}^{t-1}(1-q^{a_{i_l}}) \tag{44}$$

we can write (40) with the help of (42), (43) and (44) as

$$\begin{aligned}\Delta \leq \quad & -1 + (Q(j-1) - Q(i-1))\,(B+S) \\ & + \left(Q(i-1) - \frac{Q(h-1)}{Q_i}\right) B + Q(i-1)S \\ = & -1 + Q(j-1)(B+S) - \frac{Q(h-1)}{Q_i} B \\ \leq \quad & B + S - 1.\end{aligned} \tag{45}$$

Since $B + S \leq 1$ by Lemma 4, the desired inequality $\Delta \leq 0$ follows from (45), hence completing the proof of Theorem 2. □

6 Examples

6.1 SPS-Permutation is not optimal for general 3-level SPSs

The first example we present shows a 3-level SPS for which neither $\mathcal{S}^\rho$ nor $\mathcal{S}^*$ are optimal. In particular, both algorithms produce the same (non optimal) strategy in this case. This system is dual to the one shown in Figure (1)(a). Both the system and its tree structure are shown in figure 7, and we have $\mathbf{p} = (0.5, 0.5, 0.5, 0.5)$ and $\mathbf{c} = (5.0, 1.8, 1.0, 6.0)$. ¿From the tree structure it is easy to see that $\mathcal{S}^\rho$ and $\mathcal{S}^*$, i.e. to inspect the components in the order $(1, 2, 3, 4)$, with the expected cost of 6.9. Yet, another strategy which begins by inspecting component 2 and proceeds optimally for the residual SPSs has expected cost only of 6.8. (In particular, this strategy corresponds to the permutation $(2, 3, 1, 4)$.)

6.2 SPSpermutation is not optimal for 4-level SPSs with identical components

The next example in figure 8 shows that SPS-Permutation does not always produce optimal strategies for 4-level deep SPSs with identical components. In this example, $p_i = 0.5$ and $c_i = 1$ for all i. If we apply SPS-Permutation, subsystem 1 is inspected first till completion, then subsystem 2 is inspected, yielding an expected cost of 4.73. On the other hand, in the case of $x_1 = 1$ and $x_2 = 0$, it is advantageous to switch to subsystem 2, inspect subsystem 2 till completion and then return to the residual subsystem 1. This strategy yields an expected cost of 4.36. The latter strategy is the same strategy produced by S^*, since after x_1 and x_2 are inspected, the recalculation of the ratios would lead to switching to subsystem 2.

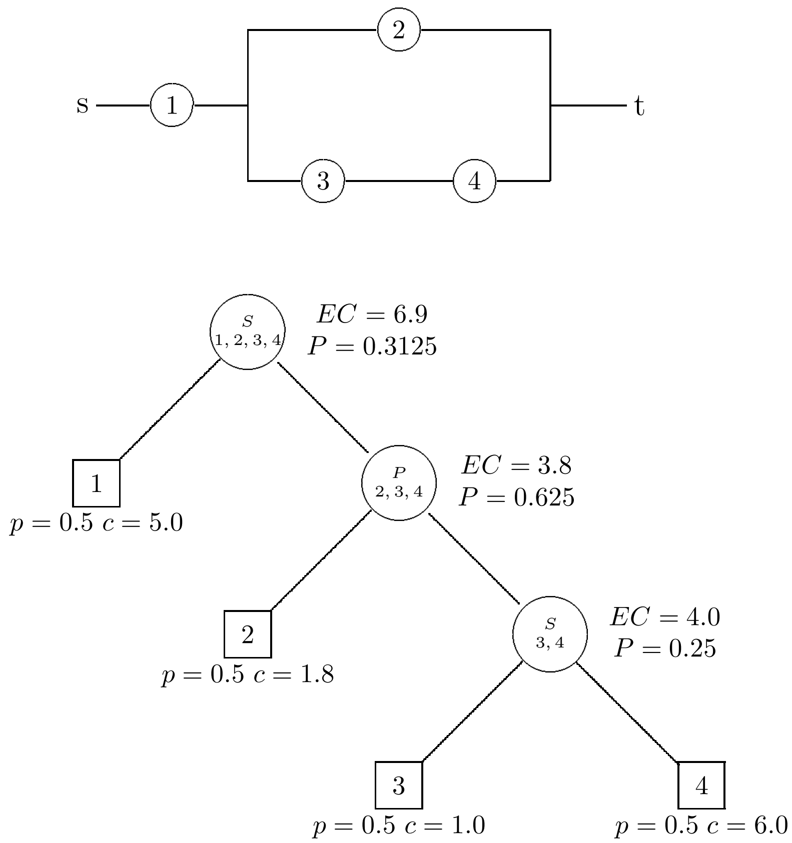

Figure 7: A 3-level SPS for which SPSpermutation does not find the optimal strategy

6.3 No optimal strategy can be described by a permutation for some SPSs

As we noted the optimal strategies for SPSs cannot always be described by permutations. The following example in figure 9 is to illustrate this fact. In this example we have $\mathbf{p} = (0.5, 0.5, 0.25, 0.5, 0.25)$ and $\mathbf{c} = (0.01, 2.0, 1.5, 1.7, 1.5)$. The unique optimal strategy for this example (found by enumerating all possible reasonable strategies) inspects component u_1 first. If component u_1 is functioning then it proceeds by inspecting component u_5 next. If component u_1 is not working then the next component to inspect is component u_3, even though component u_5 is still relevant. This shows that for this particular example one cannot describe the optimal strategy by a permutation. We can always describe the optimal strategy by a Binary Decision Tree, in a very natural way. The root of tree will be indexed by the first component to be inspected. If it works the next component to be inspected is its right child and if it fails the next component to be inspected is its left child and so on. The optimal strategy is shown in figure 9 as a Binary Decision Tree. Let us note that this

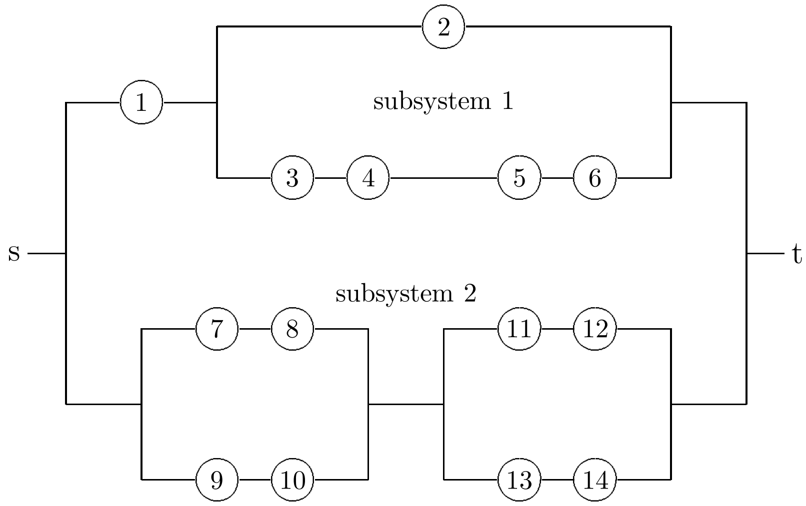

Figure 8: A 4-level SPS with identical components for which SPSpermutation does not produce an optimal strategy

is another example (see 6.1) for a 3-level SPS for which SPSpermutation does not produce an optimal strategy, yet S^* would be optimal in this case. Actually, the optimal strategy shown in figure 9 is induced by $\mathcal{S}^*$.

References

[1] B. Alidaee. Optimal ordering policy of a sequential model. *Journal of Optimization Theory and Applications*, 83:199–205, 1994.

[2] D. Angluin, L. Hellerstein, and M. Karpinski. Learning read-once formulas with queries. *Journal of the Association for Computing Machinery*, 40:185–210, 1993.

[3] L. Arseneau. Optimal testing strategies for s,t-series parallel systems. Master's thesis, Combinatorics and Optimization, University of Waterloo, 1996.

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

[5] R. Butterworth. Some reliability fault testing models. *Operations Research*, 20:335–343, 1972.

[6] W.B. Joyce. Organizations of unsuccessful R&D projects. *IEEE Transactions on Engineering Management*, EM-18(2):57–65, May 1971.

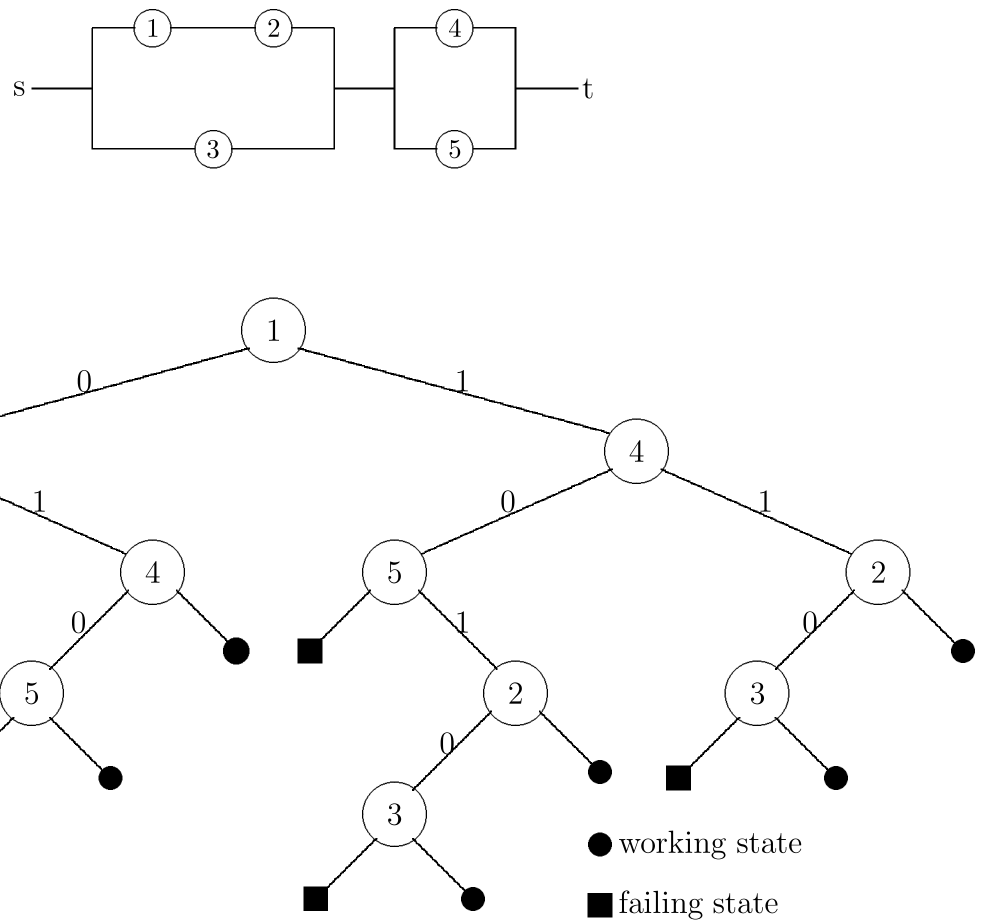

Figure 9: An SPS and its optimal strategy, which cannot be described by a permutation

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

[8] D. Mundici. Functions computed by monotone boolean formulas with no repeated variables. *Theoretical Computer Science*, 66:113–114, 1989.

[9] K.S. Natarajan. Optimizing depth-first search of AND-OR trees. Technical report, IBM T.J.Watson Research Center, Yorktown Heights, NY 10598, 1986.