Expert Systems with Applications 38 (2011) 14945–14951

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

TestAnt: An ant colony system approach to sequential testing under precedence constraints

Bülent Çatay [a], Özgür Özlük [b], Tonguç Ünlüyurt [a,*]

[a] Sabanci University, Faculty of Engineering and Natural Sciences, Tuzla 34956, Istanbul, Turkey
[b] San Francisco State University, College of Business, 94132 CA, USA

ARTICLE INFO

Keywords:
Metaheuristics
Ant colony optimization
Sequential testing
Diagnosis

ABSTRACT

We consider the problem of minimum cost sequential testing (diagnosis) of a series (or parallel) system under precedence constraints. We model the problem as a nonlinear integer program. We develop and implement an ant colony algorithm for the problem. We demonstrate the performance of this algorithm for special type of instances for which the optimal solutions can be found in polynomial time. In addition, we compare the performance of the ant colony algorithm with a branch and bound algorithm for randomly generated general instances of the problem. The ant colony algorithm is particularly effective as the problem size gets larger.

1. Introduction

In its most general setting, the sequential testing problem requires the identification of the correct state of a system consisting of a number of components with the minimum expected cost. Different states of the system could correspond to different types of failures or working conditions. The state of the system depends on the state of the components via a certain structure function. Both the state of the system and the state of the components belong to discrete sets. For instance, if we have a reliability system, the individual components and the system could be in one of the two states, failing state or working state. In another context, the system could be in one of the many failure states or in working state. In order to learn the correct state of the system, one has to learn the states of a sufficiently large subset of the components, which requires costly tests. We also assume that we have apriori probabilistic information regarding the states of the components. Then the problem is to find a strategy that minimizes total expected cost of this process. A feasible strategy, in a sense, classifies the current state of the system in a deterministic manner, by learning the states of the individual component with the minimum expected cost.

The general sequential testing problem arises in different application areas such as diagnosis problems (e.g. see Ruan, Tu, Pattipati, & Patterson-Hines, 2004; Qiu & Cox, 1993), artificial intelligence problems (e.g. see Greiner, Hayward, Jankowska, & Molloy, 2006; Wang, 2005) etc. The sequential testing problem for series–parallel systems appears as a subproblem for a resource allocation problem for reliability systems in Azaiez and Bier (2007). In addition, there are many studies in the literature that theoretically work on the problem for special structure functions and for special cases of the input data regarding probability distributions and cost. Different application areas and results related to the general sequential testing problem can be found in Ünlüyurt (2004).

In this particular study we consider the sequential testing problem of a series system under precedence constraints. A series system functions if all its components function. A feasible strategy tests the components one by one until a failing component is found or all components are tested. Consequently, a strategy for this particular system corresponds to a permutation of the components obeying the precedence constraints. In order to solve this problem, we propose an ant colony optimization (ACO) algorithm. We demonstrate the effectiveness of the proposed algorithm through an extensive experimental study. To the best of our knowledge, this is the first ACO algorithm proposed in the testing literature. Some preliminary results of this study were presented in Çatay, Özlük, and Ünlüyurt (2009).

The rest of the article is organized as follows. In Section 2, we define the problem, provide a literature review and a mathematical model of the problem. In Section 3, we describe the proposed ACO algorithm. Section 4 outlines the greedy and the branch and bound (B&B) algorithms developed to test the performance of the proposed ACO algorithm. Section 5 is devoted to the experimental study. We conclude by discussing future research directions in Section 6.

* Corresponding author. Tel.: +90 216 4839504; fax: +90 216 4839550.
E-mail address: tonguc@sabanciuniv.edu (T. Ünlüyurt).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.05.053

2. Problem definition

2.1. Problem description and literature review

We consider a series (or parallel) system where the system functions if and only if all (at least one of) the individual components function. The series (or parallel) system is a special case of k-out-of-n systems where the system functions when at least k out of its n components are functioning. A series system is an n-out-of-n system and a parallel system is a 1-out-of-n system. Since all the results for the series can be adapted to the parallel case by a dual argument, in what follows we will only consider a series system.

For instance, the system that is tested could be a newly manufactured machine that could be in working or failing state. The machine consists of independent components that are themselves in failing or working states. The machine functions if all its independent components function. In order to demonstrate that this machine is failing, one has to know that at least one component does not function. On the other hand, to conclude that the machine is functioning properly one has to know that all the components are working. Finding out whether component i works or not requires a test with a cost of c_i. By using historical data the probability that component i functions is estimated to be p_i. A testing strategy for a series system then inspects the components one by one until a failing component is found or all the components are tested. So an inspection strategy for this system is simply a permutation of all the components. The functionality of the components are independent from each other and the tests are perfect in the sense that we obtain correct information regarding the functionality of the components when they are tested. If a permutation π is used to test a series system, the expected cost of this strategy is given by

$$c_{\pi(1)} + p_{\pi(1)}c_{\pi(2)} + \cdots + p_{\pi(1)}p_{\pi(2)}\ldots p_{\pi(n-1)}c_{\pi(n)} \tag{1}$$

since component $\pi(i)$ is tested only if components $\pi(1), \pi(2), \ldots, \pi(i-1)$ are tested and all of them are functioning. With this formulation, finding an optimal sequence is easy. One just needs to test all the independent components in increasing order of $\frac{c_i}{1-p_i}$. This is an intuitive ordering since we first test the component that has less cost and high probability of failing since the testing stops when a component that is not functioning is detected. This result has been published in the literature various times in different areas and can be proved easily by an interchange argument (see e.g. Mitten, 1960).

In the case of a physical system, it may not be possible to test the components in any sequence. Certain tests can be applied only after other certain tests are performed. This could be due to the physical location of the components or technological reasons. Essentially, these constraints correspond to precedence constraints. One can describe the precedence constraints by an acyclic directed graph where the nodes of the graph correspond to the components and an arc from node i to node j means test j can be applied only after test i has been applied. We will refer to this graph as the precedence graph.

In the existence of precedence constraints there are a few analytical results for the sequential testing problem of a series system. It is proven in Reyck and Leus (2008) that when we have a general precedence graph, the problem is NP-hard. In Garey (1973) an optimal algorithm is provided when the precedence graph is a forest and each tree in the forest is either an out-tree or in-tree. In Chiu, Cox, and Sun (1999) an algorithm is proposed that is optimal for the series system when the precedence graph is a special forest where the outdegree of each node is 0 or 1. So each tree in the forest looks like a directed path. This result is a special case of the result in Garey (1973) for the series system. Both in Chiu et al. (1999) and Garey (1973), no computational results are reported. The algorithm proposed in Chiu et al. (1999) can also be used for the more general k-out-of-n systems with precedence constraints and it is shown that the proposed algorithm is optimal for certain k-out-of-n systems. Let us note that for k-out-of-n systems a testing strategy cannot be described by a permutation since the component to inspect next may depend on the results of tests that are already obtained. In general, any feasible testing strategy for k-out-of-n systems can be described by a binary tree where the nodes correspond to the components and the two outgoing arcs from a node correspond to the failing and working condition of the components. Even when there are no precedence constraints, it is not always possible to represent optimal strategies for k-out-of-n systems by a permutation.

2.2. Mathematical model

We consider a series system consisting of n components. The cost of testing component i is c_i. The tests are perfect in the sense that at a cost of c_i we learn the correct state of component i. The components can be in one of the two states; 1 corresponds to the functioning state and 0 corresponds to the failing state. The probability that component i functions is given as p_i. We assume that the functionality of the components are independent of each other. For such a system a feasible testing sequence is a permutation π of $\{1,2,\ldots,n\}$. The total expected cost of a solution π is given by $\sum_{i=1}^{n} c_{\pi(i)}\left(\prod_{j=1}^{i-1} p_{\pi(j)}\right)$. In our case, not all permutations are feasible due to the precedence constraints among the components. The precedence constraints can naturally be represented by a directed acyclic graph. If arc (i,j) exists in the precedence graph, that means component j can be tested only if component i is already tested. Let P be the set of all feasible sequences satisfying the precedence constraints. Then the problem can be formulated as follows:

$$\min_{\pi \in P} \sum_{i=1}^{n} c_{\pi(i)} \prod_{j=1}^{i-1} p_{\pi(j)} \tag{2}$$

Alternatively, the problem can be formulated as a nonlinear integer programming problem where the objective function is a nonlinear function and the constraints are linear functions:

$$\min \quad \sum_{i=1}^{n}\sum_{r=1}^{n}\left((x_{ir}c_i)\prod_{s=1}^{r-1}\sum_{j=1}^{n}(x_{js}p_j)\right) \tag{3}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_{ir} = 1, \quad \forall r \in \{1,2\ldots,n\} \tag{4}$$

$$\sum_{r=1}^{n} x_{ir} = 1, \quad \forall i \in \{1,2,\ldots,n\} \tag{5}$$

$$x_{jr} \leqslant \sum_{s=1}^{r-1} x_{is}, \ \forall (i,j) \in A \quad \text{and } r \in \{1,2\ldots,n\} \tag{6}$$

$$x_{ir} \in \{0,1\}, \quad \forall i,r \in \{1,2,\ldots,n\} \tag{7}$$

where A is the set of arcs in the precedence graph and the decision variables are defined as follows:

$$x_{ir} = \begin{cases} 1, & \text{if component } i \text{ is tested in order } r \\ 0, & \text{otherwise} \end{cases}$$

The objective function (3) represents the total expected cost of a feasible solution. The constraint (4) ensures that every component is assigned an order and the constraint (5) ensures that every order is assigned a component. Constraint (6) ensures that precedence constraints are satisfied.