# DESIGN OF OPTIMUM COMPONENT TEST PLANS WHILE CONSIDERING MULTIPLE OBJECTIVES

(ÇOK ÖLÇÜTLÜ ENİYİ BİLEŞEN SINAM PLANLARININ TASARLANMASI)

by

**Emre YAMANGİL, B.S.**

**Thesis**

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

**MASTER OF SCIENCE**

in

**INDUSTRIAL ENGINEERING**

in the

**INSTITUTE OF SCIENCE AND ENGINEERING**

of

**GALATASARAY UNIVERSITY**

May 2009

**DESIGN OF OPTIMUM COMPONENT TEST PLANS WHILE CONSIDERING MULTIPLE OBJECTIVES**

(ÇOK ÖLÇÜTLÜ ENİYİ BİLEŞEN SINAM PLANLARININ TASARLANMASI)


by

**Emre YAMANGİL, B.S.**


**Thesis**

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

**MASTER OF SCIENCE**


Date of Submission : May 8, 2009

Date of Defense Examination: May 18, 2009


Supervisor : Assoc. Prof. Dr. Orhan FEYZİOĞLU

Committee Members: Prof. Dr. İ. Kuban ALTINEL

Prof. Dr. Süleyman ÖZEKİCİ

## ACKNOWLEDGEMENTS

My deepest appreciation and thanks goes to Dr. Feyzioğlu, my mentor during this effort. It was a privilege to spend a wonderful two years of my life working under his supervision. His outstanding teaching has not only directed me to correct answers, but also taught me how to ask the right questions. I cannot be thankful enough that, through his guidance, I remembered something I had learned long ago - Given sufficient work, nothing is impossible. As he says, no effort goes to waste, try again, fail again, fail better!

During the course of my studies, I also had the chance to work with Dr. Altınel. I am forever grateful to be given an opportunity to build on his foundation, and to attend his class, combinatorial optimization.

Thanks to Dr. Özekici, this thesis has been completed to the fullest. Without his managing insight, unbelievable stochastic expertise, the optimization model would not have even been constructed.

I would like to thank Bora Çekyay, my fellow partner in the project. His stochastic formulations provided many of the bricks in this work, as well. And to Hakan Akyüz and Jbid Arsenyen, for listening to every problem I had, academic and life alike, helping to the best of their ability without the slightest hesitation.

Finally, without my sister setting me free from the throes of desperation in my junior year, my mother never letting me lose faith in myself and my father always believing in me, I couldn't even dream of accomplishing this work. Hence this work is totally yours…

I gratefully acknowledge that this research has been supported by the Turkish Scientific and Technological Research Council (TÜBİTAK)'s Grant 106M044.

08/05/2009
Emre YAMANGİL

# TABLE OF CONTENTS

# LIST OF SYMBOLS

| | |
|---|---|
| $A$ | System availability as a function of component reliabilities |
| $c_j$ | Test cost for component $j$ |
| $E[L]$ | Expected system lifetime as a function of component reliabilities |
| $F^1$ | Feasible solution index set associated with $\delta(\rho^1)$ |
| $F^2$ | Feasible solution index set associated with $\delta(\rho^2)$ |
| $f_1^i$ | Optimum solution of type I problem at iteration $i$, new column generated from $\delta(\rho^1)$ |
| $f_2^i$ | Optimum solution of type II problem at iteration $i$, new column generated from $\delta(\rho^2)$ |
| $H_0$ | Null hypothesis |
| $H_1$ | Alternative hypothesis |
| $k_j$ | Least number of components required to function for subsystem $j$ |
| $lb_j$ | Lower bound of component $j$'s failure rate |
| $m$ | Maximum number of total allowable failures |
| $m^*$ | Optimum $m$ |
| $n$ | Number of subsystems |
| $n_j$ | Number of components in subsystem $j$ |
| $N$ | Total number of failures |
| $N_j$ | Total number of failures for component $j$ |
| $P\{L > t\}$ | System reliability as a function of component reliabilities |
| $p_j(t)$ | Reliability of component $j$ at time $t$ |
| $\mathbf{p}(t)$ | Component reliability vector at time $t$ |
| $\mathbb{R}^n$ | $n$ dimensional real vectors |
| $t$ | Time |
| $t_j$ | Test time of $j^{th}$ component |
| $t_{j,m}$ | Test time of $j^{th}$ component until a total of $m$ failures occur |

| | |
|---|---|
| $t^*_{j,m}$ | Optimum test time of $j^{th}$ component until a total of $m$ failures occur |
| $ub_j$ | Upper bound on component $j$'s failure rate |
| $w^{*^i}_m$ | Optimum dual solution of $DP_i(m)$ |
| $z^*$ | $z^*_{m^*}$, minimum total test cost |
| $z^*_m$ | Minimum total test cost for a given $m$ |
| $z^{*^i}_{1,m}$ | Optimum objective value of type I problem for a given $m$ at iteration $i$ |
| $z^{*^i}_{2,m}$ | Optimum objective value of type II problem for a given $m$ at iteration $i$ |
| $\alpha$ | A given upper bound on consumer risk |
| $\beta$ | A given upper bound on producer risk |
| $\delta(\rho^1)$ | Set of feasible component failure rates for which system rejection is correct |
| $\delta(\rho^2)$ | Set of feasible component failure rates for which system acception is correct |
| $\lambda$ | Component failure rates |
| $\lambda_j$ | Component failure rate for component $j$ |
| $\lambda_m(\gamma)$ | Poisson parameter for which $\psi_m(\lambda_m(\gamma)) = \gamma$ |
| $\Lambda^1$ | $\{\lambda \in \mathbb{R}^n_+ \mid \rho(\lambda) \leq \rho^1\}$ |
| $\Lambda^2$ | $\{\lambda \in \mathbb{R}^n_+ \mid \rho(\lambda) \geq \rho^2\}$ |
| $\rho(\lambda)$ | A performance measure as a function of component reliabilities |
| $\rho^1$ | Unacceptable performance level |
| $\rho^1_R$ | Unacceptable reliability level |
| $\rho^1_M$ | Unacceptable mean time to failure level |
| $\rho^1_A$ | Unacceptable availability level |
| $\rho^2$ | Acceptable performance level |
| $\rho^2_R$ | Acceptable reliability level |
| $\rho^2_M$ | Acceptable mean time to failure level |
| $\rho^2_A$ | Acceptable availability level |
| $\phi$ | Structure function |
| $\varphi_m(\gamma)$ | Cumulative Poisson distribution with parameter $\gamma$ |
| $CDC$ | Canonical DC |
| $DC$ | Difference of two convex |

| | |
|---|---|
| $DP(m)$ | Dual linear program for a given $m$ |
| $DP_i(m)$ | Dual linear program for a given $m$ at iteration $i$ |
| $JMPM$ | Joint multiple performance measures |
| $LP$ | Linear programming |
| $P(m)$ | System-based component test problem |
| $P'(m)$ | System-based component test problem with joint multiple performance measures |
| $P''(m)$ | System-based component test problem with separate multiple performance measures |
| $P1$ | $SGP$ after exponential variable transformation |
| $P2$ | linear relaxation of $P1$ |
| $PP(m)$ | Primal linear program for a given $m$ |
| $PP_i(m)$ | Primal linear program for a given $m$ at iteration $i$ |
| $RCP$ | Reverse convex programming |
| $SGP$ | Signomial geometric programming |
| $SMPM$ | Separate multiple performance measures |

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Testing the system as a whole might be found economically infeasible or physically impossible in many cases. For instance, testing a nuclear device is currently banned by international agreements or testing a space shuttle might be found too risky because of its financial consequences. Instead, various components can be tested separately to meet some desired level of performance for the whole system, while achieving a minimal total testing cost. This problem is called system based component testing problem and is investigated in this thesis.

Although system reliability is the only considered measure in the available literature on this problem, there exist other more practical measures such as mean time to failure or availability worth to take into account. Here we extend previous studies by incorporating various performance measures separately or jointly.

The problem is formulated as a semi-infinite linear programming problem, and the optimum component test times are obtained by combining the well-known cutting plane method with the well-known column generation technique. The columns are generated by solving two different optimization subproblems which are proved to be d.c. (difference of two convex functions) programming or signomial geometric programming problems depending on the performance measures examined. These subproblems are solved to optimality by adapting an outer approximation method and by a special branch and bound technique. Several numerical examples are provided to illustrate the approach.

# RÉSUMÉ

Dans plusieurs cas, tester un système comme un tout peut sembler non-faisable économiquement ou impossible physiquement. Par exemple, tester une machine nucléaire est actuellement interdit par les accords internationaux et tester une navette spatiale peut paraître risqué étant donné les conséquences monétaires. Il est plutôt préféré de tester les divers composants pour atteindre le niveau désiré de la performance du système. Le problème investigué dans cette thèse est nommé le problème d'essai des composantes selon le système.

Même si la fiabilité du système est la seule mesure considérée dans la littérature disponible sur ce problème, il existe d'autres mesures pratiques à dévisager, comme durée moyenne de fonctionnement avant défaillance, disponibilité, etc. Les études précédentes sont élargies dans cette étude par l'intégration de ces mesures de performance séparément ou conjointement.

Le problème est formulé comme un problème de programmation linéaire semi-infinie et la durée optimale de test de composant est obtenue par la combinaison d'une technique réputée, nommé méthode des plans sécants avec une autre technique réputée, nommé méthode de génération de colonnes. Les colonnes sont engendrées en résolvant deux sous-problèmes différents d'optimisation, prouvés être des problèmes de programmation d.c. (la différence des deux fonctions convexes) ou de programmation géométrique signomiale, suivant la mesure de performance examinée. Ces sous-problèmes sont résolus à l'optimalité par l'adaptation d'une technique d'approximation par l'extérieure et par une méthode spéciale de séparation et d'évaluation progressive. Plusieurs exemples numériques sont fournis afin d'illustrer l'approche.

# ÖZET

Dizgeyi bir bütün halinde sınamak olanaksız veya katlanılamayacak bir maliyete olabilir. Örneğin nükleer silahların denenmesi uluslararası anlaşmalar çerçevesinde sınırlandırılmıştır. Benzer şekilde bir uzay mekiğini sınamanın mali sonuçları oldukça belirsizdir. Bu gibi durumlarda bileşenler, dizgenin bütünü için öngörülmüş başarım ölçütlerini sağlayacak şekilde ve olabilecek en düşük maliyette ayrı ayrı denenebilirler. Bu problem dizge tabanlı bileşen sınamı olarak bilinmektedir ve bu tezin konusunu oluşturmaktadır.

Probleme ilişkin yazında sadece sistem güvenilirliği dikkate alınmaktadır. Ancak beklenen yaşam süresi veya kullanılırlık gibi ölçütler uygulamada daha kullanışlı bulunabilir. Bu çalışma farklı başarım ölçütlerini hem ayrık hem de bir arada kullanarak varolan diğer çalışmaları genellemektedir.

Problemin yarı-sonsuz doğrusal programlama modeli oluşturulmuş ve eniyi bileşen sınam süreleri kesme-düzlem yöntemi ile sütun üretme yordamının bir araya getirilmesiyle hesaplanmıştır. Sütunlar, ele alınan başarım ölçütleri doğrultusunda d.c. (dışbükey fonksiyonların farkı) programlama veya genel geometrik programlama problemi oldukları kanıtlanmış iki eniyileme alt probleminin çözülmesi ile üretilmiştir. Bu alt problemlerin çözümünde dıştan yaklaşıklama veya dal-sınır yöntemlerinden faydalanılmıştır. Yaklaşımın ayrıntılarını gözler önüne sermek için çok sayıda sayısal örnek hazırlanmıştır.

# 1 INTRODUCTION

Increasing product complexity, challenging environment and competition in global market have led to an increase in performance demands from consumers. For example many military devices carry out critical missions in a growingly hostile territory where a system failure can result in operator injury, damage to property, and a significant economic loss.

When a system failure occurs, no matter how benign, its impact is felt. For example, even a screw fallen from a chair may have consequences. A small injury or at least a need to repair the chair before utilizing its full use thereafter. However, a disfunctional flap in an airplane may result in a plane crash or an inevitable landing.

Although it is almost impossible to fully avoid system failure in the long run, it is still important to reduce its occurrence probability. Instead of trying to eliminate a system failure, system can be tested for whether it satisfies a certain performance measure or not, at some pre-determined level before undertaking its mission. These tests can provide information about a system's capacity to fulfill its requirements. For example, using these statistics, one can either design a system to meet its objective, avoid a system breakdown using preventive maintenance at critical phases or know the limits of a system before assigning it a mission. Then the question becomes "How do we design a test model to examine the system against a desired performance measure?"

There are two well-known approaches for testing a system. First, one can either test the system as a whole, simulating the circumstances of mission's environment as best as possible and examine how does the system act in this simulation. Second, one can conduct individual component tests for the prediction and verification of the system performance. Although it is cheaper and less dangerous to examine the components of a system rather than the whole, the first approach provides better insights into the system's capacity. However the cost of these tests can be overwhelming, preventing its usage in many situations. In this thesis, we are concerned with the alternative test approach that combines the strong features of the two, *the identification of cost-efficient component test plans to demonstrate a desired performance measures for a system*.

Component testing is carried out when it is economically infeasible or physically impossible to test the system as a whole. For example, testing a nuclear device is currently banned by international agreements or testing a space shuttle might be found too risky because of its financial consequences. These are extreme but typical examples where one needs to attain a certain performance level without testing the system. Instead the test is done on various components to meet a desired performance measure for the whole system, while achieving a minimal testing cost.

This approach, known as *system-based component testing*, drew a lot of attention in the past three decades. It amounts to find a component testing plan that assures some system performance within predetermined limits with minimum total testing cost. Researchers considered different system topologies, extended the problem from independent to dependent environments, introduced systems designed to accomplish multi-phased missions. However, system reliability is regarded as a single performance criterion in most of the available literature. In this work, we consider the case in which setting a system's expected lifetime or availability are more practical than determining its reliability. Furthermore, we formulate the situation where system is expected to satisfy a set of performance measures rather than only satisfying single criterion at pre-determined levels. We formulate the multiple performance measure system-based component testing problem in two modeling approaches.

Within the existing definition and formulation of the problem, there are two main difficulties which make this problem hard solve to optimality. First, even it is a linear programming problem and has finitely many variables, it also has infinitely many constraints. This type of problems are called semi-infinite linear programming problems. Approaches to solve this problem typically include a column generation scheme to generate a finite subset of the constraints to satisfy optimality conditions.

Instead of generating rows and solving the problem from scratch every time a row is generated, the applied methodology works on the dual of the original problem using the "revised simplex" method to save from the valuable computational time.

Second, our main problem requires solving two separate subproblems to optimality within the column generation process. These subproblems have compact but non-convex solution sets. Hence any standard convex programming approach is not useful. Meanwhile, it can be shown that the constraints of these subproblems can be explicitly expressed as a *difference of two convex functions* or as a *ratio of two posynomials* whenever it is most convenient. It is well-known that exploiting these special structures is very convenient for solving the resulting non-convex optimization problems globally. In this thesis, we adapt some existing deterministic procedures in the literature and embed within the column generation scheme to solve the system based component testing problem to optimality.

The remaining parts of this thesis are composed of six sections. In Section 2, we present a brief introduction to basic reliability theory, and formulate system reliability, expected system lifetime and system availability for various system topologies. Then the semi-infinite linear programming formulation of system-based component testing problem is provided along with the solution method in Section 3. After we give detailed information on procedures used in the solution of the subproblems in Section 4, we illustrate the theoretical work with numerical examples in Section 5. Finally, Section 6 includes concluding remarks.

# 2 SYSTEM PERFORMANCE MEASURES

In this chapter, we give a brief introduction to the basic reliability theory and derive reliability, mean time to failure and availability performance measures for serially connected redundant, standby redundant and $k$-out-of-$n$ subsystems. The random variables representing the state of the components are assumed to be independent.

## 2.1 BASIC RELIABILITY THEORY

Consider a system of $n$ components, and suppose that each component is either functioning or failed. To indicate if the $i$th component is functioning or not, we define the indicator binary variable $x_i$ as,

$$x_j = \begin{cases} 1 & \text{if the } i\text{th component is functioning} \\ 0 & \text{if the } i\text{th component has failed} \end{cases} . \tag{2.1}$$

Similarly, the binary variable $\phi$ is indicating the state of the system:

$$\phi = \begin{cases} 1 & \text{if the system is in functioning state} \\ 0 & \text{if the system is in failed state} \end{cases} . \tag{2.2}$$

Further, we assume that the state of the system is a function of the state of the components. If we denote the state vector of components with $x = (x_1, x_2, \ldots, x_n)$. Then the state of the system regarding the component state vector can be given by

$$\phi = \phi(x). \tag{2.3}$$

Here $\phi(x)$ is called the *structure function* of the system.

**Example 2.1 (Series System)**

A system consisting of $n$ components that is functioning if and only if all of its components are functioning is called a series system. The structure function of a series system can be given as follows

$$\begin{aligned} \phi(x) &= \min(x_1, x_2, \ldots, x_n) \\ &= \prod_{i=1}^{n} x_i. \end{aligned} \tag{2.4}$$

A series structure is illustrated by the reliability block diagram in Figure 2.1.

Figure 2.1 Series system

**Example 2.2 (Parallel System)**

A system that is functioning if and only if at least one of its components is functioning is called a parallel system. Its structure function can be given by,

$$\begin{aligned} \phi(x) &= \max(x_1, x_2, \ldots, x_n) \\ &= 1 - \prod_{i=1}^{n}(1 - x_i). \end{aligned} \tag{2.5}$$

The corresponding reliability block diagram is given in Figure 2.2.

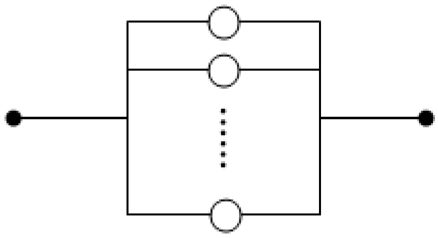

Figure 2.2 Parallel system

**Example 2.3 (*k*-out-of-*n* System)**

A system which is functioning if and only if at least $k$ out of $n$ of its components are functioning, is called a $k$-out-of-$n$ system. In terms of comparability, a series system is a $n$-out-of-$n$ system and a parallel system is a 1-out-of-$n$ system. Hence one can say that $k$-out-of-$n$ systems is a generalization of both series and parallel systems. The structure function of a $k$-out-of-$n$ system can be given as follows,

$$\phi(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i \geq k \\ 0 & \text{if } \sum_{i=1}^{n} x_i < k \end{cases} . \tag{2.6}$$

The reliability block diagram of a 2-out-of-3 structure is provided in Figure 2.3 for illustration.

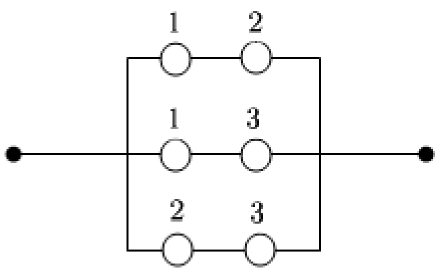


Figure 2.3 Structure of a 2-out-of-3 system

**Definition 2.1 (Monotone System)**

A system is said to be monotone if

i. Its structure function $\phi$ is nondecreasing in each argument, and
ii. $\phi(1) = 1$ and $\phi(0) = 0$.

The first argument says that a system cannot deteriorate by improving the state of a component, namely by replacing a failed component with a functioning one. The second argument says that the system is in functioning state if all the components are functioning at the moment and in failed state if all the components are failed.

Let $(l_i, x) = (x_1, \dots, x_{i-1}, l, x_{i+1}, \dots, x_n)$ denote the state of $i$th component $l \in \{0, 1\}$.

**Definition 2.2 (Coherent System)**

A system is said to be coherent if

i. Its structure function $\phi$ is nondecreasing in each argument, and
ii. Each component is relevant, i.e., there exists at least one vector $(\cdot_i, x)$ such that $\phi(1_i, x) = 1$ and $\phi(0_i, x) = 0$.

Detailed information on these definitions can be found in Barlow and Proschan [1].

**Definition 2.3 (Minimal Path Set)**

A state vector $x$ is called a minimal path vector if

i. $\phi(x) = 1$ and
ii. $\phi(y) = 0$ for all $y < x$.

If $x$ is a minimal path vector, then the set $A = \{i : x_i = 1\}$ is called a *minimal path set*. In other words, a minimal path set is a minimal set of components whose functioning ensures the functioning of the system.

Let $P = \{P_1, \dots, P_s\}$ denote the minimal path sets of a given system. We define $\alpha_j(x)$, the indicator function of the $j$th minimal path set, by

$$\alpha_j(x) = \begin{cases} 1, & \text{if all the components of } P_j \text{ are functioning} \\ 0, & \text{otherwise} \end{cases}$$

$$= \prod_{i \in P_j} x_i.$$

It follows that the system will function if all the components of at least one minimal path set are functioning; that is $\alpha_j(x) = 1$ for some $j$. Hence,

$$\phi(x) = \begin{cases} 1, & \text{if } \alpha_j(x) = 1 \text{ for some } j \\ 0, & \text{if } \alpha_j(x) = 0 \text{ for all } j \end{cases}$$

or equivalently

$$\begin{aligned} \phi(x) &= \max_j \alpha_j(x) \\ &= \max_j \prod_{i \in P_j} x_i. \end{aligned} \tag{2.7}$$

**Definition 2.4 (Minimal Cut Set)**

A state vector $\mathbf{x}$ is called a minimal cut vector if

i. $\phi(x) = 0$ and

ii. $\phi(y) = 1$ for all $y > x$.

If $\mathbf{x}$ is a minimal cut vector, then the set $C = \{i : x_i = 0\}$ is called a *minimal cut set*. In other words, a minimal cut set is a minimal set of components whose failure ensures the failure of the system.

Let $C = \{C_1, \ldots, C_r\}$ denote the minimal cut sets of a given system. We define $\beta_j(x)$, the indicator function of the $j$th minimal cut set, by

$$\beta_j(x) = \begin{cases} 1, & \text{if at least one component of } C_j \text{ is functioning} \\ 0, & \text{if all the components of } C_j \text{ are not functioning} \end{cases}$$

$$= \max_{i \in C_j} x_i.$$

Since a system is not functioning if and only if all the components of at least one minimal cut set are not functioning, it follows that

$$\begin{aligned} \phi(x) &= \prod_{j=1}^{k} \beta_j(x) \\ &= \prod_{j=1}^{k} \max_{i \in C_j} x_i. \end{aligned} \tag{2.8}$$

For example let's consider the bridge system.

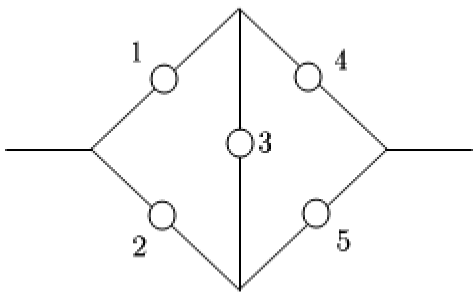


Figure 2.4 The bridge system

The system structure is as illustrated in Figure 2.4. The minimal path sets are $\{1,4\}$, $\{1,3,5\}$, $\{2,5\}$, and $\{2,3,4\}$. Hence by equation (2.7), its structure function may be expressed as

$$\phi(x) = \max\{x_1x_4, x_1x_3x_5, x_2x_5, x_2x_3x_4\}$$

$$= 1 - (1 - x_1x_4)(1 - x_1x_3x_5)(1 - x_2x_5)(1 - x_2x_3x_4).$$

$$\begin{aligned} &= x_2x_3x_4 + x_2x_5 - x_2^2x_5x_3x_4 + x_1x_3x_5 - x_1x_3^2x_5x_2x_4 \\ &\quad + x_1x_3^2x_5^2x_2^2x_4 - x1x_4^2x_2x_3 - x_1x_4x_2x_5 - x_1^2x_4^2x_3^2x_5^2x_2^2 \\ &\quad + x_1x_4^2x_2^2x_5x_3 - x_1^2x_4x_3x_5 + x_1^2x_4^2x_3^2x_5x_2 + x_1^2x_4x_3x_5^2x_2 \\ &\quad - x_1x_3x_5^2x_2. \end{aligned}$$

The minimal cut sets of the bridge system are $\{1, 2\}$, $\{1, 3, 5\}$, $\{4, 5\}$, and $\{2, 3, 4\}$. Hence, from equation (2.8), the structure function of a bridge system can be given as follows

$$\phi(x) = \max(x_1, x_2)\max(x_1, x_3, x_5)\max(x_4, x_5)\max(x_2, x_3, x_4)$$

$$= [1 - (1 - x_1)(1 - x_2)][1 - (1 - x_1)(1 - x_3)(1 - x_5)]$$

$$\times [1 - (1 - x_4)(1 - x_5)][1 - (1 - x_2)(1 - x_3)(1 - x_4)].$$

$$\begin{aligned} &= x_2x_3x_4 + x_2x_5 - x_2^2x_5x_3x_4 + x_1x_3x_5 - x_1x_3^2x_5x_2x_4 \\ &\quad + x_1x_3^2x_5^2x_2^2x_4 - x1x_4^2x_2x_3 - x_1x_4x_2x_5 - x_1^2x_4^2x_3^2x_5^2x_2^2 \\ &\quad + x_1x_4^2x_2^2x_5x_3 - x_1^2x_4x_3x_5 + x_1^2x_4^2x_3^2x_5x_2 + x_1^2x_4x_3x_5^2x_2 \\ &\quad - x_1x_3x_5^2x_2. \end{aligned}$$

## 2.2 SYSTEM RELIABILITY

We assign each component a random variable $X_i(t)$, denoting the state of the $i$th component at time $t$, such that,

$$P\{X_i(t) = 1\} = p_i(t) = 1 - P\{X_i(t) = 0\}. \tag{2.9}$$

Here the value $p_i(t)$ is called the *reliability* of the $i$th component at time $t$. If we define $r$ by

$$r = P\{\phi(X(t)) = 1\}, \quad \text{where} \quad X(t) = (X_1(t), \ldots, X_n(t)), \tag{2.10}$$

then $r$ is called the *reliability* of the system. By assuming that the random variables $X_i(t)$ are independent than each other, we can express $r$ as a function of component reliabilities $p_i(t),\ i = 1, \ldots, n$. That is,

$$r = r(\mathbf{p}(t)), \quad \text{where} \quad \mathbf{p}(t) = (p_1(t), \ldots, p_n(t)). \tag{2.11}$$

This function is known as the *reliability function*. We now provide system reliability function for some coherent and noncoherent structures with components having exponential lifetimes.

### 2.2.1 Series Systems

Assuming that the components fail exponentially with rate $\lambda_j$, the reliability function of the series system of $n$ independent components is given by

$$\begin{aligned} r(\mathbf{p}(t)) &= P\{\phi(X(t)) = 1\} \\ &= P\{X_i(t) = 1 \text{ for all } i = 1, \ldots, n\} \\ &= \prod_{i=1}^{n} e^{-\lambda_i t}. \end{aligned} \tag{2.12}$$

### 2.2.2 Serial Connection of Redundant Subsystems

The reliability function of serially connected $n$ redundant subsystems with each subsystem having $n_j$, $j = 1, \ldots, n$ independent identical components with exponential failure rate $\lambda_j$, is given by

$$\begin{aligned} r(\mathbf{p}(t)) &= \prod_{j=1}^{n} P\{\phi(X^j(t)) = 1\} \\ &= \prod_{j=1}^{n} P\{X_i^j(t) = 0 \text{ for some } i = 1, \ldots, n_j\} \\ &= \prod_{j=1}^{n} (1 - P\{X_i^j(t) = 0 \text{ for all } i = 1, \ldots, n_j\}) \\ &= \prod_{j=1}^{n} (1 - (1 - e^{-\lambda_j t})^{n_j}). \end{aligned} \tag{2.13}$$

### 2.2.3 Serial Connection of *k*-out-of-*n* Subsystems

Consider $n$ serially connected $k$-out-of-$n$ subsystems. Subsystem $j$ consists of $n_j$ identical components with exponential failure rates $\lambda_j$, and requires the functionality of $k_j$ of its components to survive. Reliability of this system is given below.

$$\begin{aligned} r(\mathbf{p}(t)) &= \prod_{j=1}^{n} P\{\phi(X^j(t)) = 1\} \\ &= \prod_{j=1}^{n} P\{\sum_{i=1}^{n} X_i^j(t) \geq k_j\} \\ &= \prod_{j=1}^{n} \sum_{i=k_j}^{n} \binom{n}{i} (e^{-\lambda_j t})^i (1 - e^{-\lambda_j t})^{n-i}. \end{aligned} \tag{2.14}$$

### 2.2.4 Serial Connection of Standby Redundant Subsystems

It is well known that a standby redundant system is not a coherent system. Consider serially connected $n$ standby redundant subsystems. Each subsystem has $n_j$, $j = 1, \ldots, n$ independent identical components with exponential failure rate $\lambda_j$. The reliability function of such a system is given by,

$$\begin{aligned} r(\mathbf{p}(t)) &= \prod_{j=1}^{n} P\{\phi(X^j(t)) = 1\} \\ &= \prod_{j=1}^{n} \sum_{r=0}^{n_j - 1} \frac{e^{-\lambda_j t}(\lambda_j t)^r}{r!}. \end{aligned} \tag{2.15}$$

## 2.3 SYSTEM MEAN TIME TO FAILURE

### 2.3.1 Coherent Systems

Consider a system that consists of $n$ subsystems and each subsystem is composed of identical components. Each component of subsystem $j$ fails independently and exponentially with rate $\lambda_j$. If $L$ is the random variable for the system lifetime, then $E[L]$ denotes the expected lifetime or Mean Time To Failure (MTTF) of the system and $r(\mathbf{p}(t)) = P\{\phi(X(t)) = 1\} = P\{L > t\}$ denotes the system reliability expressed as a function of component reliabilities, where $\mathbf{p}(t) = (\mathbf{p}_1(t), \ldots, \mathbf{p}_n(t))$ $= (e^{-\lambda_1 t}, \ldots, e^{-\lambda_n t})$. Then the relation between the system's expected lifetime and reliability can be given as

$$E[L] = \int_0^{+\infty} P\{L > t\}\, dt = \int_0^{+\infty} \phi(x) dt \tag{2.16}$$

Let $S = \{x \in I^m : \phi(x) = 1\} \subset I^m$ and $F = \{x \in I^m : \phi(x) = 0\} \subset I^m$ denote the set of all path and cut vectors respectively. For any state $x \in I^m$, let $C_1(x) = \{k : x_k = 1\}$

denote the set of functioning components and $C_0(x) = \{k : x_k = 0\}$ denote the set of failed components. Now we can equivalently represent the structure function (2.7) as follows

$$\phi(x) = 1 - \prod_{z \in P} (1 - \prod_{i \in C_1(z)} x_i) = \sum_{y \in S} ( \prod_{i \in C_1(y)} x_i)( \prod_{j \in C_0(y)} (1 - x_j)). \qquad (2.17)$$

By substituting

$$\prod_{j \in C_0(y)} (1 - x_j) = \sum_{k=0}^{|C_0(y)|} (-1)^k \sum_{j_1, j_2, \ldots, j_k \in C_0(y)} \prod_{n=1}^{k} x_{j_n}, \qquad (2.18)$$

we can equally represent (2.17) by

$$\phi(x) = \sum_{y \in S} \sum_{k=0}^{|C_0(y)|} (-1)^k \sum_{j_1, j_2, \ldots, j_k \in C_0(y)} \prod_{\substack{i \in C_1(y), \\ n=1,\ldots,k}} x_i x_{j_n}, \qquad (2.19)$$

where $i \neq j_n$ for all $i \in C_1(y)$ and $j_n \in C_0(y)$, and $j_1, j_2, \ldots, j_k \in C_0(y)$ represents $k$ different combinations of the elements of $C_0(y)$. Now assuming that

i. components fail exponentially and independent than each other,
ii. failed components are repaired immediately and separately after a system break down [2],

we can give the explicit reliability function of a coherent system [3] by

$$P\{L > t\} = \sum_{y \in S} \sum_{k=0}^{|C_0(y)|} (-1)^k \sum_{j_1, j_2, \ldots, j_k \in C_0(y)} e^{-\left( \sum_{\substack{i \in C_1(y) \\ l=1,\ldots,k}} (\lambda_i + \lambda_{j_n}) \right)}. \qquad (2.20)$$

by taking the definite integral of reliability function (2.16), we can express MTTF of a coherent system as follows

$$E[L] = \sum_{y \in S} \sum_{k=0}^{|C_0(y)|} (-1)^k \sum_{j_1, j_2, \ldots, j_k \in C_0(y)} \left( \sum_{\substack{i \in C_1(y) \\ l=1,\ldots,k}} (\lambda_i + \lambda_{j_l}) \right)^{-1} . \quad (2.21)$$

### 2.3.2 Series Systems

The reliability and expected system lifetime of a series system consisting of $n$ components can be given as

$$P\{L > t\} = e^{-(\lambda_1 + \ldots + \lambda_n)t}, \quad (2.22)$$

$$E[L] = \frac{1}{\lambda_1 + \cdots + \lambda_n}. \quad (2.23)$$

### 2.3.3 Serial Connection of Redundant Subsystems

Assuming that each subsystem consists of $n_j$ identical components, the reliability and expected system lifetime of a serial connection of redundant subsystems can be given as

$$\begin{aligned} P\{L > t\} = &\sum_{r_1=1}^{n_1} \cdots \sum_{r_n=1}^{n_n} \sum_{s_1=r_1}^{n_1} \cdots \sum_{s_n=r_n}^{n_n} \binom{n_1}{s_1} \binom{s_1}{r_1} \cdots \\ &\times \binom{n_n}{s_n} \binom{s_n}{r_n} (-1)^{s-r} e^{-(s_1\lambda_1 + \ldots + s_n\lambda_n)t}, \end{aligned} \quad (2.24)$$

$$E[L] = \sum_{r_1=1}^{n_1} \cdots \sum_{r_n=1}^{n_n} \sum_{s_1=r_1}^{n_1} \cdots \sum_{s_n=r_n}^{n_n} \binom{n_1}{s_1} \binom{s_1}{r_1} \cdots \times \binom{n_n}{s_n} \binom{s_n}{r_n} (-1)^{s-r} \left( \frac{1}{s_1\lambda_1 + \cdots + s_n\lambda_n} \right). \tag{2.25}$$

Here $s = s_1 + \ldots + s_n$ and $r = r_1 + \ldots + r_n$.

### 2.3.4 Serial Connection of *k*-out-of-*n* Subsystems

MTTF of $k$-out-of-$n$ systems is analyzed in [4]. The author assumes that all lifetimes and repair times are independent and exponentially distributed, there are enough repairmen for all failed components and replacement for a component starts immediately after failure. Assuming each subsystem consists of $n_j$ identical components, expected lifetime of the serial connection of $k$-out-of-$n$ subsystems can be derived from the reliability function such as given below.

$$P\{L > t\} = \sum_{r_1=k_1}^{n_1} \cdots \sum_{r_n=k_n}^{n_n} \sum_{s_1=r_1}^{n_1} \cdots \sum_{s_n=r_n}^{n_n} \binom{n_1}{s_1} \binom{s_1}{r_1} \cdots \times \binom{n_n}{s_n} \binom{s_n}{r_n} (-1)^{s-r} e^{-(s_1\lambda_1 + \ldots + s_n\lambda_n)t}, \tag{2.26}$$

$$E[L] = \sum_{r_1=k_1}^{n_1} \cdots \sum_{r_n=k_n}^{n_n} \sum_{s_1=r_1}^{n_1} \cdots \sum_{s_n=r_n}^{n_n} \binom{n_1}{s_1} \binom{s_1}{r_1} \cdots \times \binom{n_n}{s_n} \binom{s_n}{r_n} (-1)^{s-r} \left( \frac{1}{s_1\lambda_1 + \cdots + s_n\lambda_n} \right). \tag{2.27}$$

### 2.3.5 Serial Connection of Standby Redundant Subsystems

It is well-known that the structure function of this kind of systems is not coherent. Therefore, the results in the previous sections are not applicable. However, by assuming that all components have exponential lifetimes and the components in each subsystem are identical [5], system reliability can be expressed explicitly. Suppose that there are $m$ subsystems and subsystem $k$ consists of $n_k$ identical components with exponential failure rates $\lambda_k$. Then one can explicitly express system reliability (2.15), and hence expected system lifetime as [3]

$$P\{L > t\} = \prod_{k=1}^{m} \sum_{r_k=0}^{n_k-1} \frac{e^{-\lambda_k t}(\lambda_k t)^{r_k}}{r_k!}$$
$$= \sum_{r_1=0}^{n_1-1} \cdots \sum_{r_n=0}^{n_n-1} \frac{\lambda_1^{r_1} \ldots \lambda_n^{r_n}}{r_1! \ldots r_n!} e^{(\lambda_1+\ldots+\lambda_m)t} \times t^{r_1+\ldots+r_m} \tag{2.28}$$

and

$$E[L] = \sum_{r_1=0}^{n_1-1} \cdots \sum_{r_n=0}^{n_n-1} \frac{\lambda_1^{r_1} \ldots \lambda_n^{r_n}(r_1 + \ldots + r_n)!}{r_1! \ldots r_n!(\lambda_1 + \ldots + \lambda_n)^{r_1+\ldots+r_n+1}}. \tag{2.29}$$

## 2.4 SYSTEM AVAILABILITY

Assuming that all component lifetimes and repair times are exponential, availability can be determined using Markovian analysis. The states of the corresponding Markov process will depend on the system structure and we need to find limiting distribution. In this section availability functions of coherent systems and non-coherent systems are derived by Çekyay and Özekici [3] using the Markovian analysis.

### 2.4.1 Coherent Systems

Let $F_S = \{x \in F : (1_i, x) \in S \text{ for some } i = 1, \ldots, n\}$ for any $x \in I^n$. It is assumed that the repair starts when the system enters some state $x \in F_S$, which takes an exponentially distributed amount of time with some rate $\mu_x > 0$. After the repairing process, all of the components are in functioning state. Let $\overline{1} = (1, \ldots, 1)$ represent the perfect state. Now we can say that the states of the system follow a Markov process with state space $E = S \cup F_S$ since all lifetimes and repair times are exponential. We need to find the limiting distribution to express system availability in terms of component failure rates. Therefore we need to solve the system of linear equations

$$\begin{aligned}
\pi_{\overline{1}} \sum_{k=1}^{n} \lambda_k &= \sum_{x \in F_S} \pi_x \mu_x, \\
\pi_x \sum_{j \in C_1(x)} \lambda_j &= \sum_{j \in C_0(x)} \pi_{(1_j, x)} \lambda_j, \quad \forall x \in S \setminus \{\overline{1}\}, \\
\pi_x \mu_x &= \sum_{\substack{j \in C_0(x), \\ (1_j, x) \in S}} \pi_{(1_j, x)} \lambda_j, \quad \forall x \in F_S, \\
\sum_{x \in S} \pi_x + \sum_{x \in F_S} \pi_x &= 1,
\end{aligned} \tag{2.30}$$

Then we can give the system availability in terms of functioning states as follows

$$A = \sum_{x \in S} \pi_x. \tag{2.31}$$

Note that since $\mu_x > 0$ for all $x \in F_S$ and $\lambda_k > 0$ for all $k = 1, \ldots, n$, the embedded Markov chain is irreducible with non-null recurrent states. Hence, the system of linear equations (2.30) has a unique solution [3].

As the serial connection of series subsystems, passive redundant subsystems and $k$-out-of-$n$ subsystems falls under coherent system category, their availability can be formulated using (2.30) and (2.31).

### 2.4.2 Series Systems

Let's consider the series system of $n$ components and assume that all component $j$ fails exponentially with failure rate $\lambda_j$. Let the state space represent the number of available components in each subsystem such that

$$E = \{(i_1, \ldots, i_n) : i_j = 0, 1,\ j = 1, \ldots, n \text{ where } i_j = 0 \text{ for only one } j\}.$$

System starts in the initial state $\overline{1} = (1, \ldots, 1)$ and it will be repaired whenever one of its components $j$ enters a failure state with $i_j = 0$. We can give the failure states in terms of a failed component $j$ by

$$F_j = \{x \in E : x_j = 0 \text{ and } x_i = 1,\ \forall i \neq j\}$$

and all of the failure states by

$$F_S = \bigcup_{j=1,\ldots,n} F_j.$$

It takes an exponentially distributed amount of time with some rate $\mu_x > 0$, for all $x \in F_S$, to repair the system upon entering the failure state. It is clear that states of the system follow a Markov process with state space $E = S \cup F_S$ since all lifetimes and repair times are exponentially distributed and the limiting distribution can be found by solving the system of linear equations

$$\begin{aligned}
\pi_{\overline{n}} \sum_{i=1}^{n} \lambda_i &= \sum_{x \in F_S} \pi_x \mu_x, \\
\pi_x \sum_{i=1}^{n} x_i \lambda_i &= \sum_{j \in O(x)} \pi_{(1_j^+, x)} (1 + x_j) \lambda_j, \quad x \in S \setminus \{\overline{1}\}, \\
\pi_x \mu_x &= \pi_{(1_j^+, x)} \lambda_j, \quad x \in F_j,\ j = 1, \ldots, n, \\
\sum_{x \in S} \pi_x + \sum_{x \in F_S} \pi_x &= 1,
\end{aligned} \tag{2.32}$$

where $O(x) = \{j : x_j < 1\}$ and $(1_j^+, x) = (x_1, \ldots, x_{j-1}, x_j + 1, x_{j+1}, \ldots, x_n)$ for $x \in E$.

Availability function can be formulated with (2.31), using the solution of (2.32)

### 2.4.3 Serial Connection of Redundant Subsystems

Let's consider the serial connection of $n$ redundant subsystems each of which is consisting $n_j$ identical components and assume that all components of subsystem $j$ fail exponentially with failure rate $\lambda_j$. Let the state space represent the number of available components in each subsystem such that

$$E = \{(i_1, \ldots, i_n) : i_j = 0, \ldots, n_j, j = 1, \ldots, n \text{ where } i_j = 0 \text{ for only one } j\}.$$

System starts in the initial state $\overline{n} = (n_1, \ldots, n_n)$ and it will be repaired whenever one of its subsystems $j$ enters a failure state with $i_j = 0$. We can give the failure states in terms of a failed subsystem $j$ by

$$F_j = \{x \in E : x_j = 0 \text{ and } x_i \geq 1, \ \forall i \neq j\}.$$

Repairing takes an exponentially distributed amount of time with some rate $\mu_x > 0$ whenever system enters a state $x \in F_S = \cup_{j=1,\ldots,n} F_j$. We can determine the limiting distribution using Markov process with state space $E = S \cup F_S$ by solving the system of linear equations

$$
\begin{aligned}
\pi_{\overline{n}} \sum_{i=1}^{n} n_i \lambda_i &= \sum_{x \in F_S} \pi_x \mu_x, \\
\pi_x \sum_{i=1}^{n} x_i \lambda_i &= \sum_{j \in O(x)} \pi_{(1_j^+, x)} (1 + x_j) \lambda_j, \quad x \in S \setminus \{\overline{n}\}, \\
\pi_x \mu_x &= \pi_{(1_j^+, x)} \lambda_j, \quad x \in F_j,\ j = 1, \ldots, n, \\
\sum_{x \in S} \pi_x + \sum_{x \in F_S} \pi_x &= 1.
\end{aligned}
\tag{2.33}
$$

where $O(x) = \{j : x_j < 1\}$. We need to obtain the solution of (2.33), before formulating availability function using (2.31).

### 2.4.4 Serial Connection of *k*-out-of-*n* Subsystems

Let's consider the serial connection of $n$ $k$-out-of-$n$ subsystems and assume that all components of subsystem $j$ fail exponentially with failure rate $\lambda_j$. Let the state space represent the number of available components in each subsystem such that

$$E = \{(i_1, \ldots, i_n) : i_j = k_j - 1, \ldots, n_j, j = 1, \ldots, n \text{ where } i_j = k_j - 1 \text{ for only one } j\}.$$

System starts in the initial state $\overline{n} = (n_1, \ldots, n_n)$ and it will be repaired whenever one of its subsystems $j$ enters a failure state with $i_j = k_j - 1$. We can give the failure states in terms of a failed subsystem $j$ by

$$F_j = \{x \in E : x_j = k_j - 1 \text{ and } x_i \geq k_i,\ \forall i \neq j\}.$$

If system enters a state belonging to the set $F_S = \cup_{j=1,\ldots,n} F_j$ repairing phase is triggered and it takes an exponentially distributed amount of time with some rate $\mu_x > 0$, for all $x \in F_S$, to repair the system thereafter. Again the states of the system follow a Markov process with state space $E = S \cup F_S$ and the limiting distribution can be found by solving the system of linear equations

$$
\begin{aligned}
\pi_{\overline{n}} \sum_{i=1}^{n} n_i \lambda_i &= \sum_{x \in F_S} \pi_x \mu_x, \\
\pi_x \sum_{i=1}^{n} x_i \lambda_i &= \sum_{j \in O(x)} \pi_{(1_j^+, x)} (1 + x_j) \lambda_j, \quad x \in S \setminus \{\overline{n}\}, \\
\pi_x \mu_x &= \pi_{(1_j^+, x)} k_j \lambda_j, \quad x \in F_j,\ j = 1, \ldots, n, \\
\sum_{x \in S} \pi_x + \sum_{x \in F_S} \pi_x &= 1.
\end{aligned}
\tag{2.34}
$$

where $O(x) = \{j : x_j < k_j\}$. After obtaining the solution of (2.34), we can formulate availability function using (2.31).

### 2.4.5 Serial Connection of Standby Redundant Subsystems

We inspect a system of $n$ serially connected standby redundant subsystems each having $n_j$ identical components. Let the state space is represented as follows

$$E = \{(i_1, \ldots, i_n) : i_j = 0, 1, \ldots, n_j, j = 1, \ldots, n \text{ where } i_j = 0 \text{ for only one } j\}$$

and the failure states in terms of subsystem $j$ fails

$$F_j = \{x \in E : x_j = 0 \text{ and } x_i \geq 1 \text{ for all } i \neq j\}.$$

It is obvious that the system fails whenever it enters a state $x \in F_S = \cup_{j=1,\ldots,n} F_j$ and it takes an exponentially distributed time $\mu_x > 0$ to get the system fully operational. We can formulate availability function using (2.31) upon solving the set of linear equations

$$\begin{aligned}
\pi_{\overline{n}} \sum_{i=1}^{n} \lambda_i &= \sum_{x \in F_S} \pi_x \mu_x, \\
\pi_x \sum_{i=1}^{n} \lambda_i &= \sum_{j \in O(x)} \pi_{(1_j^+, x)} \lambda_j, \quad x \in S \setminus \{\overline{n}\}, \\
\pi_x \mu_x &= \pi_{(1_j^+, x)} \lambda_j, \quad x \in F_j, j = 1, \ldots, n, \\
\sum_{x \in S} \pi_x + \sum_{x \in F_S} \pi_x &= 1,
\end{aligned} \tag{2.35}$$

In the following chapters, we are going to assume that $\mu_x = 1 \ \forall x \in F_S$ without loss of generality

# 3 SYSTEM-BASED COMPONENT TESTING PROBLEM

Let $\rho(\lambda)$ denote some performance measure for a system with $n$ components and component failure rate vector $\lambda = (\lambda_1, \ldots, \lambda_n)$. Then the component testing problem can be stated in terms of the hypothesis testing problem

$$H_0 : \rho(\lambda) \leq \rho_1, \quad H_1 : \rho(\lambda) \geq \rho_2 \tag{3.1}$$

where $\rho_1$ and $\rho_2$ indicate unacceptable and acceptable system performance levels respectively ($\rho_1 < \rho_2$ by definition). It is widely known that two errors can occur when testing a hypothesis. Type I error is known as rejecting $H_0$ when $H_0$ is true and type II error is rejecting $H_1$ when $H_1$ is true.

Our focus is to devise a model that minimizes the total component testing cost, while assuring type I and type II error probabilities to be less than desirable levels. Let $\alpha$ and $\beta$ denote the upper bounds on type I and type II errors. Let also $t_j$ denotes the test time, $c_j$ the non-negative test cost and $N_j$ the number of failures of component $j$, and $m$ the upper bound on the total number of component failures. Then the system-based component testing problem can be formulated as

$$\min \quad \sum\nolimits_{j=1}^{n} c_j t_j, \tag{3.2}$$

s.t.

$$P\,[\,\text{Reject } H_0 | H_0 \text{ is true}\,] \leq \alpha, \tag{3.3}$$

$$P\,[\,\text{Reject } H_1 | H_1 \text{ is true}\,] \leq \beta. \tag{3.4}$$

$$t_j \geq 0, \; j = 1, \ldots, n \tag{3.5}$$

This problem was first mentioned by Gal [6]. In this work, the author proposes to minimize total component testing cost, $\sum_{j=1}^{k} c_j t_j$, for a system where a certain unacceptable reliability level, $R_0$, needs to be demonstrated at $1-\alpha$ confidence interval (3.3). He also assumes exponential life distributions for components. Mazumdar [7] extends Gal's model by also considering an acceptable system reliability level $R_1$ that needs to be demonstrated at a specified confidence, $1-\beta$. This boils down to including constraint (3.4) in his model. Further, instead of accepting a system if and only if there are no component failures during the test as Gal did ($m = 0$), Mazumdar proposes to accept a system if the total number of component failures, $\sum_{j=1}^{k} N_j$, is less than a threshold value, say $\sum_{j=1}^{k} N_j \leq m$, and reject otherwise. This rule is referred as "sum rule". Note that this is a generalization of Gal's rule, which he considers the case $m = 0$ only. Easterling *et al.* [8] give a justification for using the sum rule for a series system.

Using the sum rule, Mazumdar provides an algorithm to compute optimum number of component failures, $m^*$, which minimizes the total component testing cost, also meets the unacceptable and acceptable reliability levels. He gives two numerical examples; a series system and a series system with redundant subsystems with the assumption of component lifetimes are independently exponentially distributed.

In their respective formulations, both Gal [6] and Mazumdar [7] show that for a series system, the optimum component test times are independent of component test costs and are identical. They both assume that no prior information is available about component reliabilities. Altınel [9] considers the case where some prior information on component reliabilities exists as a mean of setting upper bounds on component failure rates. With the use of this prior information, he shows that the optimum component test times are not identical, and the use of such information also leads to reduced total test cost. He also develops a procedure to compute optimum component test times. Altınel and Özekici [10] extend these results to a dynamically changing environment where these upper bounds on component failure rates change with respect to time. For modeling this concept, they introduce arbitrary distributions for component failure rates which can be approximated by distributions that have piecewise constant failure rates. This is accomplished through a dynamic environmental process that modulates component failure rates. Since the failure rate of each component is constant during any environment, components still fail exponentially. However, the failure rates change whenever the state of the environment changes. Therefore, lifetime distributions are not necessarily exponential, but the piecewise constant structure of the failure rates is exploited to obtain tractable expressions for the reliability function at the expense of an enlarged set of failure rates. A major assumption of the previous formulations of system-based component test problem is the independence of component failure rates, which is a rather restrictive and unrealistic assumption for most cases. Altınel and Özekici [11] use an interesting model of stochastic component dependence introduced by Çınlar and Özekici [12] and generalize these results further in order to compute optimum component test times with dependent components. In all of the above models the system is assumed nonrepairable, hence no maintenance is done throughout the mission time. Altınel *et al.* [13] introduce missions that involve a sequence of stages where a maintenance operation is carried away in the beginning of each stage. This maintenance operation consists of checking the device and replacing failed components with identical ones so that the functioning state of the system at the start of each stage is preserved. Altınel *et al.* [14] analyze the case where there is a given set of missions and the device can be assigned randomly to these missions. Feyzioğlu *et al.* [15-16] extends the variety of systems considered by including $k$-out-of-$n$ and standby redundant subsystems. They also show that serial connection of different subsystems can also be modeled for both single and multi-phased missions.

## 3.1 FORMULATION

Let us reconsider the system-based component testing problem given in (3.2) - (3.4) with some performance measure $\rho(\lambda)$ needs to be demonstrated at $1-\alpha$ and $1-\beta$ levels.

$$\min \quad \sum_{j=1}^{n} c_j t_j, \tag{3.6}$$

s.t.

$$P\,[\text{ Reject } H_0|H_0 \text{ is true }] = P\left[\sum_{j=1}^{n} N_j \leq m \,|\, \rho(\lambda) \leq \rho^1\right] \leq \alpha, \tag{3.7}$$

$$P\,[\text{ Reject } H_1|H_1 \text{ is true }] = P\left[\sum_{j=1}^{n} N_j > m \,|\, \rho(\lambda) \geq \rho^2\right] \leq \beta. \tag{3.8}$$

$$t_j \geq 0, \; j = 1, \ldots, n \tag{3.9}$$

We denote $\Lambda^1 = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho(\lambda) \leq \rho^1\right\}$ and $\Lambda^2 = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho(\lambda) \geq \rho^2\right\}$, as the feasible failure rate sets satisfying constraints (3.7) and (3.8), respectively. We also assume that some prior information as lower and upper bounds on each component's failure rate exists and is obtained without additional costs. With this information, we can rewrite the feasible failure rate sets $\Lambda^1$ and $\Lambda^2$ as

$$\delta(\rho^1) = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho(\lambda) \leq \rho^1, \, lb_j \leq \lambda_j \leq ub_j \;\; j = 1, \ldots, n\right\} \tag{3.10}$$

and

$$\delta(\rho^2) = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho(\lambda) \geq \rho^2, \, lb_j \leq \lambda_j \leq ub_j \;\; j = 1, \ldots, n\right\} \tag{3.11}$$

respectively. Assuming that all components fail exponentially, $N_j$ is Poisson distributed with mean $\lambda_j t_j$, and $\sum_{j=1}^{n} N_j$ is Poisson distributed with mean $\sum_{j=1}^{n} \lambda_j t_j$. If $\delta(\rho^1)$ and $\delta(\rho^2)$, are nonempty, there exists at least one solution to the system (3.6) - (3.9). The probability constraints (3.7) and (3.8) are surely guaranteed for all feasible $\lambda$ vectors if they are modified as

$$\max_{\lambda \in \delta(\rho^l)} \; P\left[\sum_{j=1}^{n} N_j \leq m\right] \leq \alpha \tag{3.12}$$

and

$$\min_{\lambda\in\delta(\rho^2)} \; P\left[\sum\nolimits_{j=1}^{n} N_j \leq m\right] \geq 1-\beta. \tag{3.13}$$

Let $Y$ be a Poisson random variable with parameter $y$ and $\varphi_m(y) = P\left[Y \leq m\right]$ denotes the cumulative Poisson distribution function. Then $P[\sum_{j=1}^{n} N_j \leq m] = \varphi_m(\sum_{j=1}^{n} \lambda_j t_j)$ is the system acceptance probability and we can rearrange the probability constraints (3.12) and (3.13) as

$$\max_{\lambda\in\delta(\rho^1)} \; \varphi_m\left(\sum\nolimits_{j=1}^{n} \lambda_j t_j\right) \leq \alpha \tag{3.14}$$

and

$$\min_{\lambda\in\delta(\rho^2)} \; \varphi_m\left(\sum\nolimits_{j=1}^{n} \lambda_j t_j\right) \geq 1-\beta. \tag{3.15}$$

As $\varphi_m\left(y\right)$ is strictly decreasing and continuous for a given value of $m$, it is also invertible with respect to $y$. Let $\lambda_{\gamma,m}$ be the Poisson parameter value for which $\varphi_m\left(\lambda_{\gamma,m}\right) = \gamma$. Therefore (3.14) and (3.15) can be further arranged as

$$\min_{\lambda\in\delta(\rho^1)} \; \sum\nolimits_{j=1}^{n} \lambda_j t_j \geq \lambda_{\alpha,m} \tag{3.16}$$

and

$$\max_{\lambda\in\delta(\rho^2)} \; \sum\nolimits_{j=1}^{n} \lambda_j t_j \leq \lambda_{1-\beta,m}. \tag{3.17}$$

With this inversion, the problem given in (3.6) - (3.9) can be reformulated as follows:

$P(m)$ :

$$\min \quad \sum_{j=1}^{n} c_j t_{j,m}, \tag{3.18}$$

s.t.

$$\min\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \,|\, \lambda \in \delta(\rho^1)\right\} \geq \lambda_{\alpha,m}, \tag{3.19}$$

$$\max\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \,|\, \lambda \in \delta(\rho^2)\right\} \leq \lambda_{1-\beta,m}, \tag{3.20}$$

$$t_{j,m} \geq 0 \quad j = \{1, \ldots, n\}. \tag{3.21}$$

The solution of $P(m)$ is denoted by $\left\{t^*_{j,m} : j = \{1, \ldots, n\}\right\}$. These are the component test times which yields the minimum total component testing cost for a given value of $m$, and $z^*_m$ is the associated total test cost. Then the minimum total test cost is $z^* = z^*_{m^*} = \min\{z^*_m : m = 1, 2, \ldots\}$ and it is obtained by solving $P(m)$ parametrically with respect to $m$. Then the optimum component test times $\left\{t^*_j : j = \{1, \ldots, n\}\right\}$ is the optimal solution of $P(m^*)$. In the following chapters, optimization problems on the left hand side of (3.19) and (3.20) are referred as type I and type II problems.

## 3.2 FORMULATION WITH MULTIPLE PERFORMANCE MEASURES

Let us reconsider the system-based component testing problem in the following *multiple performance measure* formulations. Let $Z = \{R, M, A\}$ denote the set of performance measures, $R$ denotes system reliability, $M$ denotes system expected lifetime and $A$ denotes system availability, respectively. Furthermore let $\rho_R(\lambda) = P\{L > t\}$, $\rho_M(\lambda) = E[L]$ and $\rho_A(\lambda) = A$. Let $\rho^1_l$ and $\rho^2_l$, denote unacceptable and acceptable performance levels for each performance measure $l \in Z$, respectively. The multiple performance measures formulation can be handled using two approaches. Either we can formulate one type I and one type II problem with $|Z|$ performance constraints each, or we can separately formulate $|Z|$ type I and type II problems each one capturing the unacceptable and acceptable performance levels of one performance measure, respectively.

### 3.2.1 Joint Multiple Performance Measures Formulation

Now let's consider $P(m)$ given in (3.18) - (3.21). One can formulate the multiple performance measure system-based component testing problem as a Joint Multiple Performance Measure (abbreviated as JMPM so forth) formulation given below,

$P'(m)$ :

$$\min \quad \sum_{j=1}^{n} c_j t_{j,m}, \tag{3.22}$$

s.t.

$$\min\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \,|\, \lambda \in \delta'(\rho_R^1, \rho_M^1, \rho_A^1)\}\right\} \geq \lambda_{\alpha,m}, \tag{3.23}$$

$$\max\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \,|\, \lambda \in \delta'(\rho_R^2, \rho_M^2, \rho_A^2)\}\right\} \leq \lambda_{1-\beta,m}, \tag{3.24}$$

$$t_{j,m} \geq 0 \;\; j = \{1, \ldots, n\}, \tag{3.25}$$

where

$$\delta'(\rho_R^1, \rho_M^1, \rho_A^1) = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho_l(\lambda) \leq \rho_l^1, \forall l \in Z,\, lb_j \leq \lambda_j \leq ub_j, j = 1, \ldots, n\right\}, \tag{3.26}$$

$$\delta'(\rho_R^2, \rho_M^2, \rho_A^2) = \left\{\lambda \in \mathbb{R}_+^n \,|\, \rho_l(\lambda) \geq \rho_l^2, \forall l \in Z, lb_j \leq \lambda_j \leq ub_j, j = 1, \ldots, n\right\}, \tag{3.27}$$

denote the set of feasible failure rate vectors, respectively.

### 3.2.2 Separate Multiple Performance Measures Formulation

Instead of taking the intersection of $|Z|$ performance measures, one can convert $P(m)$ given in (3.18) - (3.21) to a multiple performance measure test problem considering each performance measure separately. Hence forming $|Z|$ type I and type II problems and generating columns by solving each of these optimization problems separately.

$P''(m)$ :

$$\min \quad \sum_{j=1}^{n} c_j t_{j,m}, \tag{3.28}$$

s.t.

$$\min\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \mid \lambda \in \delta(\rho_l^1)\right\} \geq \lambda_{\alpha,m}, \quad \forall l \in Z, \tag{3.29}$$

$$\max\left\{\sum_{j=1}^{n} \lambda_j t_{j,m} \mid \lambda \in \delta(\rho_l^2)\right\} \leq \lambda_{1-\beta,m}, \quad \forall l \in Z, \tag{3.30}$$

$$t_{j,m} \geq 0 \;\; j = \{1, \ldots, n\}. \tag{3.31}$$

We will call $P''(m)$ as the Separate Multiple Performance Measures (abbreviated as SMPM so forth) formulation. It is especially useful when the JMPM formulation becomes difficult to solve.

## 3.3 SOLUTION PROCEDURE

The optimization problem $P(m)$ given in (3.18) - (3.21) has finitely many variables and infinitely many constraints. In other words, it is a semi-infinite linear programming problem. We now give a brief description of a solution procedure which solve this type of problems and which is also based on earlier works of Altınel [17-18].

With a computational point of view, we assume that $\delta(\rho^1)$ and $\delta(\rho^2)$ are finite sets, in other words $f_i^1 \in \delta(\rho^1)$ and $f_i^2 \in \delta_l(\rho_l^2)$ for every $i \in F^1$ and $i \in F^2$, respectively. This discretization strategy is also used to solve other semi-infinite linear programming problems effectively [19]. Let $PP(m)$ and $DP(m)$ denote the primal and dual problems associated with $P(m)$. Then

$PP(m)$ :

$$\min \quad \sum_{j=1}^{n} c_j t_j \,, \tag{3.32}$$

s.t.

$$\sum_{j=1}^{n} f_{i,j}^1 t_{j,m} \geq \lambda_{\alpha,m} \qquad i \in F^1 \quad \text{(dual variable } \pi_i^1), \tag{3.33}$$

$$\sum_{j=1}^{n} f_{i,j}^2 t_{j,m} \leq \lambda_{1-\beta,m} \qquad i \in F^2 \quad \text{(dual variable } \pi_i^2), \tag{3.34}$$

$$t_{j,m} \geq 0 \;\; j = \{1, \ldots, n\}. \tag{3.35}$$

$DP(m)$ :

$$\begin{aligned} \text{max} \quad & \sum_{i \in F^1} \lambda_{\alpha,m} \pi_i^1 - \sum_{i \in F^2} \lambda_{1-\beta,m} \pi_i^2 \,, & (3.36) \\ \text{s.t.} \quad & & \\ & \sum_{i \in F^1} f_{i,j}^1 \pi_i^1 - \sum_{i \in F^2} f_{i,j}^2 \pi_i^2 \le c_j \qquad j = \{1, \ldots, n\} & (3.37) \\ & \pi_i^1 \ge 0 \;\; i \in F^1, \; \pi_i^2 \ge 0 \;\; i \in F^2 \,. & (3.38) \end{aligned}$$

Here, if $F^1$ and $F^2$ are finite and chosen so that the component test times which solve $P(m)$ to optimality are in the feasible solution set of $PP(m)$, then solving $DP(m)$ solves $P(m)$. A close investigation shows that it is more convenient to work on $DP(m)$ since the number of columns can be substantially larger than the number of rows. Moreover, $DP(m)$ is always feasible for all $m \in \mathbb{N}$ given that test costs $c_j$ are non-negative in (3.32).

The formulation of $DP(m)$ allows us also to introduce nonnegative slack variables for each row, hence $(n \times n)$ identity matrix as a basis for $DP(m)$. The solution algorithm proposed here is based on the general cutting plane method for convex programs combined with the column generation technique. Starting with empty $F^1$ and $F^2$ or equivalently unconstrained $PP(m)$, we generate new linear inequalities and solving $PP(m)$ until an $\epsilon-$optimal solution (more precisely, a solution arbitrarily close to the optimal solution) is obtained. Since adding a new constraint to $PP(m)$ is equivalent to adding a new variable to its dual $DP(m)$, instead of solving $PP(m)$ from scratch, we solve $DP(m)$ by using the revised simplex algorithm. The basis is updated by pivoting on the new generated column to be added to the constraint matrix of $DP(m)$.

Since $DP(m)$ is always feasible, the procedure can stop only in two possible cases. Either we detect the unboundedness of $DP(m)$ or we solve it to optimality. It is well known that the unboundedness of the dual problem means the infeasibility of the primal problem. In other words, $PP(m)$ is infeasible, and in turn, the original problem $P(m)$ is infeasible due to the fact that the current constraint set with indices $F^1$ and $F^2$ is a relaxation of the feasible set of $P(m)$.

When the above procedure does not stop, at least one column is generated and added to the constraint matrix of $DP(m)$, and then the optimum solution of updated $DP(m)$ is found. By the linear programming duality, the optimum dual solution of $DP(m)$ with this new column set is the optimum solution of $PP(m)$ with the new constraint set. The above procedure stops after computing an $\epsilon-$optimal solution of $DP(m)$. This implies that the current optimal solution is also optimal for any larger column sets containing the current column set as a subset. Hence the dual of an $\epsilon-$optimal solution of $DP(m)$ is an $\epsilon-$optimal solution of $PP(m)$; it is in fact an $\epsilon-$optimal solution of the semi-infinite linear programming problem $P(m)$.

Let us consider $DP(m)$ for a given set of columns with indices $F^1$ and $F^2$, and assume that it is bounded. Then, the simplex algorithm stops if and only if the reduced cost $z_j - h_j \geq 0$ for all nonbasic columns of $DP(m)$, or equivalently $\min\{z_j - h_j : \text{ for every nonbasic } j\} \geq 0$ (here $h$ is used to avoid confusion with the unit test cost vector $c$). We observe that the index of a nonbasic column can be either in $F^1$ or in $F^2$. Moreover, $h_j = \lambda_m(\alpha)$ for all $j \in F^1$, and $h_j = -\lambda_m(1-\beta)$ for all $j \in F^2$. Then, by denoting an optimal solution of $DP(m)$ by $w_m^*$ and using the fact that $t_m^* = w_m^*$, we can write the stopping condition of the simplex algorithm as

$$(\min_{i \in F^1} f_{1,l}^i t_m^* \geq \lambda_m(\alpha) \text{ AND } \min_{i \in F^2} f_{2,l}^i t_m^* \geq -\lambda_m(1-\beta)) \tag{3.39}$$

or equivalently as

$$(\min_{i \in F^1} f_{1,l}^i t_m^* \geq \lambda_m(\alpha) \text{ AND } \max_{i \in F^2} f_{2,l}^i t_m^* \leq \lambda_m(1-\beta)). \tag{3.40}$$

If we slightly modify this stopping condition to consider all possible nonbasic columns, which are to be generated from the feasible failure rate sets $\delta(\rho^1)$ and $\delta(\rho^2)$, then the simplex algorithm stops if and only if type I and type II constraints given as inequalities (3.19) and (3.20) in the original formulation of $P(m)$ are satisfied, or equivalently, if and only if

$$(\min_{\lambda \in \delta(\rho^1)} t_m^* \lambda \geq \lambda_m(\alpha) \text{ AND } \max_{\lambda \in \delta(\rho^2)} t_m^* \lambda \leq \lambda_m(1-\beta)). \tag{3.41}$$

This condition requires the solution of two optimization problems in $\lambda$, whose objective coefficients are the current optimal dual solution of $P(m)$. Any optimum solution of these two optimization problems which violate its related inequality (3.41) generates a new column to be added to the constraint matrix of $DP(m)$, which is a new cut for $PP(m)$. This procedure is more formally illustrated in the next algorithm proposed by Altınel [17].

**Algorithm 3.1 Column generation algorithm to solve *P(m)*.**

---

*Step* 0. Input $\rho^1$, $\rho^2$, $\lambda_m(\alpha)$, $\lambda_m(1-\beta)$, $c$, $lb$, $ub$; Initialize dual solution $w_m^{*^1} \leftarrow 0$, inverse basis $B_1^{-1} \leftarrow I_{(n\times e)(n\times e)}$, dual objective $z_m^* \leftarrow 0$ and iteration counter $i \leftarrow 1$;

*Step* 1. $z_{1,m}^{*^i} \leftarrow \min\{w_m^{*^i}\lambda : \lambda \in \delta(\rho^1)\}$ and call the optimum solution $f_1^i$;

$z_{2,m}^{*^i} \leftarrow \max\{w_m^{*^i}\lambda : \lambda \in \delta(\rho^2)\}$ and call the optimum solution $f_2^i$;

*Step* 2. if $(z_{1,m}^{*^i} \geq \lambda_m(\alpha)$ AND $z_{2,m}^{*^i} \leq \lambda_m(1-\beta))$

STOP, $t_m^* = w_m^{*^i}$ are the optimum component test times and $z_m^* = z_{D,m}^{*^i}$ is the minimum total test cost for this value of $m$;

else

UPDATE $B_i^{-1}$ with $f_1^i$, $f_2^i$ as two new columns;

UPDATE dual solution $w_m^{*^i}$;

$i \leftarrow i+1$;

end if

*Step* 3. Solve $DP_i(m)$ with inverse basis $B_i^{-1}$;

*Step* 4. if $DP_i(m)$ is BOUNDED, go to *Step* 1;

else, STOP and output "INFEASIBLE $m$";

end if

---

We must also search for the optimum value of $m$ to compute the optimum test times. As it is explained in Altınel [17], $z_m^*$ is approximately a convex function of $m$. Consequently, it is possible to search for $m^*$, the value of $m$ for which $z_m^* < z_{m+1}^*$ holds for the first time, starting from $m = 0$ by using the column generation algorithm for computing $z_m^*$ values. We can assume that $z_m^* = \infty$ for any value of $m$, $DP(m)$ is unbounded, or equivalently $PP(m)$ is infeasible. Although this does not always guarantee the optimum solution, stopping at the first $m$ that minimize $PP(m)$ turns out to be a good heuristic rule in practice.

# 4 SOLUTION METHODS TO SOLVE SUBPROBLEMS

Within the general solution framework given above, it is required to optimally solve type I and type II problems explicitly. Most of the time, these subproblems are nonconvex. Feyzioğlu *et al.* [16] proves that the reliability functions given in (2.12) - (2.15) log-concave functions. This means after taking the natural logarithm of the reliability constraint, type I problem becomes a linear reverse convex optimization problem and type II problem becomes a convex minimization problem. We also exploit this structure and solve subproblems involving reliability constraints with an outer approximation procedure proposed by Horst and Tuy [20]. MTTF functions given in (2.21), (2.23), (2.25), (2.27) and (2.29) and availability functions constructed from the limiting distributions (2.30), (2.32), (2.33), (2.34) and (2.35) can be also transformed to a difference of two convex functions with algebraic manipulations. But the resulting optimization problems are rather complicated to solve after these derivations. By rearranging the terms appropriately, MTTF functions can be reformulated as a ratio of two posynomials [3]. To illustrate, consider the serial connection of one 2-out-of-3 and one 3-out-of-4 subsystems:

$$\begin{aligned} E[L] &= \frac{12}{2\lambda_1 + 3\lambda_2} - \frac{9}{2\lambda_1 + 4\lambda_2} - \frac{8}{3\lambda_1 + 3\lambda_2} + \frac{6}{3\lambda_1 + 4\lambda_2} \\ &= \frac{30\lambda_1^3 + 175\lambda_1^2\lambda_2 + 245\lambda_1\lambda_2^2 + 84\lambda_2^3}{36\lambda_1^4 + 210\lambda_1^3\lambda_2 + 450\lambda_1^2\lambda_2^2 + 420\lambda_1\lambda_2^3 + 144\lambda_2^4}. \end{aligned}$$

Therefore, we can equivalently represent system expected lifetime as

$$E[L] = \frac{\sum_{u=1}^{U} s_u \prod_{j=1}^{n} \lambda_j^{a_{uj}}}{\sum_{v=1}^{V} r_v \prod_{j=1}^{n} \lambda_j^{b_{vj}}}, \tag{4.1}$$

where $s_u$ and $r_v$ are positive coefficients, $a_{uj}$ and $b_{vj}$ are integer constant exponents. Let $\rho_M^1$ denote the unacceptable level of MTTF and similarly $\rho_M^2$ denote the acceptable level of MTTF. Using general representation function given in (4.1), optimization problems in (3.19) and (3.20) can be restated as

$$\begin{array}{ll} \min & \sum_{j=1}^{n} t_j \lambda_j \\ \text{s.t.} & \sum_{u=1}^{U} s_u \prod_{j=1}^{n} \lambda_j^{a_{uj}} - \rho_M^1 \sum_{v=1}^{V} r_v \prod_{j=1}^{n} \lambda_j^{b_{vj}} \leq 0 \\ & lb_j \leq \lambda_j \leq ub_j \quad j = 1, \dots, n \end{array},$$

$$\begin{array}{ll} \max & \sum_{j=1}^{n} t_j \lambda_j \\ \text{s.t.} & \sum_{u=1}^{U} s_u \prod_{j=1}^{n} \lambda_j^{a_{uj}} - \rho_M^2 \sum_{v=1}^{V} r_v \prod_{j=1}^{n} \lambda_j^{b_{vj}} \geq 0 \\ & lb_j \leq \lambda_j \leq ub_j \quad j = 1, \dots, n \end{array}.$$

Both type I and type II problems are now Signomial Geometric Programming problems, which can be solved globally using a branch and bound scheme described in Shen *et al.* [21]. We describe the details of this algorithm in section 4.2.

This case also applies if availability is considered. Let us consider the serial connection of a single component and a 2-out-of-3 subsystem. Using the limiting distribution formulated in (2.30) for coherent systems, we need to solve the following system of linear equations

$$\begin{aligned} \pi_{13}(\lambda_1 + 3\lambda_2) &= \pi_{03} + \pi_{02} + \pi_{11}, \\ \pi_{12}(\lambda_1 + 2\lambda_2) &= 3\lambda_2 \pi_{13}, \\ \pi_{03} &= \lambda_1 \pi_{13}, \\ \pi_{02} &= \lambda_1 \pi_{12}, \\ \pi_{11} &= 2\lambda_2 \pi_{12}, \\ \pi_{13} + \pi_{12} + \pi_{03} + \pi_{02} + \pi_{11} &= 1. \end{aligned}$$

After obtaining the unique solution to the above system, we can formulate the availability function from (2.31) as follows

$$A = \frac{5\lambda_2 + \lambda_1}{5\lambda_2 + 5\lambda_1\lambda_2 + 6\lambda_2^2 + \lambda_1 + \lambda_1^2}.$$

It is clear that this availability function has the same structure with (4.1). Hence availability sub problems can be handled using the same procedure.

## 4.1 DC PROGRAMMING

### 4.1.1 DC Functions

Convexity is a nice property of functions which, unfortunately, is not preserved even under such simple algebraic operations as scalar multiplication or lower envelope. Now we give a brief definition to the *d.c. structure* (also called the *complementary convex structure*) which is the common underlying mathematical structure of virtually all nonconvex optimization problems [22].

Let $S$ be a convex set in $\mathbb{R}^n$. We say that a function is *d.c.* on $S$ if it can be expressed as the difference of two convex functions on $S$, i.e. $f(x) = f_1(x) - f_2(x)$, where $f_1(x),\ f_2(x)$ are convex functions on $S$.

An inequality of the form $f(x) \leq 0$, where the function $f(x)$ is convex, is called a convex inequality (because the set of all $x$ satisfying this inequality is a convex set). If $f(x)$ is concave, then the inequality is called complementary convex or reverse convex because its solution set is the complement of a convex set. Thus a reverse convex inequality is of the form $f(x) \geq 0$, where $f(x)$ is convex. If $f(x)$ is a *d.c.* function then the inequality $f(x) \leq 0$ is called a *d.c.* inequality. The following proposition shows the wide range of applicability of *d.c.* functions. Let $C^k(\mathbb{R}^n)$ denotes the class of functions on $\mathbb{R}^n$ continuously differentiable up to order $k$.

**Proposition 4.1**

*Every function $f \in C^2(\mathbb{R}^n)$ is d.c. on any compact convex set $S \subset \mathbb{R}^n$* [22].

However, it is not very easy to find the *d.c.* representation of a given function. Introductory information on the *d.c.* decomposition of basic composite functions, separable functions and polynomials can be found in the work of Horst and Tuy [20]. A *d.c.* set $S' \subset \mathbb{R}^n$ can be represented as $S' = \{x : g(x) \leq 0,\ h(x) \geq 0\}$ where both functions $g,\ h : \mathbb{R}^n \to \mathbb{R}$ are convex. In other words, $S' = D \setminus C$ where $D = \{x : g(x) \leq 0\}$ and $C = \{x : h(x) \geq 0\}$.

### 4.1.2 Canonical DC Programming

A global optimization problem is called a *d.c. programming problem* if it has the form,

$$\text{(DC)}: \begin{cases} \text{Minimize} & g_0(x) \\ \text{subject to} & g_j(x) \leq 0, \quad j = 1, \ldots, m, \\ & x \in C, \end{cases} \tag{4.2}$$

where $C \subset \mathbb{R}^n$ is convex and all functions $g_j$ are *d.c.* on $C$, which is usually given by a set of convex inequalities. By introducing at most two additional variables, every *d.c.* programming problem can be transformed into an equivalent *canonical d.c. programming* (CDC) problem

$$\text{(CDC)}: \begin{cases} \text{Minimize} & f(x) = cx \\ \text{subject to} & g(x) \leq 0 \\ & h(x) \geq 0, \end{cases} \tag{4.3}$$

where $c \in \mathbb{R}^n$, and where $h$ and $g : \mathbb{R}^n \to \mathbb{R}$ are real valued convex functions on $\mathbb{R}^n$.

Let $D = \{x : g(x) \leq 0\}$ and $C = \{x : h(x) \geq 0\}$. If an optimal solution $w$ of the convex program $\min\{cx : g(x) \leq 0\}$ satisfies $h(w) \geq 0$, then the problem is solved. However the reverse convex constraint is not essential in the problem. Therefore, without loss of generality we may assume that there exists a point $w$ satisfying

$$w \in \text{int}D \cap \text{int}C, \quad f(x) > f(w) \quad \forall x \in D \setminus C \tag{4.4}$$

The next important property is an immediate consequence of this assumption.

**Proposition 4.2 (boundary property)**
*Every global optimal solution lies on* $D \cap \partial C$ [22].

*Proof* Let $z^0$ be any feasible solution. If $z^0 \notin \partial C$, then the line segment $[w; z^0]$ meets $\partial C$ at a point $z^1 = (1-\lambda)w + \lambda z^0$ such that $0 < \lambda < 1$. By convexity we have from (4.4), $f(z^1) \leq (1-\lambda)f(w) + \lambda f(z^0) < f(z^0)$, so $z^1$ is a better feasible solution than $z^0$.

Problem CDC is said to be regular if the feasible set $S = D \setminus \text{int}C$ is robust or, which amounts to,

$$D \setminus \text{int}C = \text{cl}(D \setminus C). \tag{4.5}$$

**Theorem 4.1 (global optimality condition)**
*In order that a feasible solution* $\bar{x}$ *to CDC be global optimal it is necessary that* [22]

$$\{x \in D : f(x) \leq f(\bar{x})\} \subset C. \tag{4.6}$$

This condition is also sufficient if the problem is regular. To exploit this optimality criterion, it is convenient to introduce the next concept. Given $\epsilon > 0$, a vector $\bar{x}$ is said to be $\epsilon$-approximate optimal solution to CDC if

$$\bar{x} \in D, \quad h(\bar{x}) \geq -\epsilon \tag{4.7}$$

$$f(\bar{x}) \leq \min\{f(x) : x \in D, h(x) \geq 0\}. \tag{4.8}$$

Clearly as $\epsilon \to 0$, any accumulation point of a sequence $\{\bar{x}^k\}$ of $\epsilon$-approximate optimal solutions to CDC yields an exact global optimal solution. Therefore, in practice one should be satisfied with an $\epsilon$-approximate optimal solution for $\epsilon$ sufficiently small. Denote $C_\epsilon = \{x | h(x) \leq -\epsilon\},\ D(\gamma) = \{x \in D | f(x) \leq \gamma\}$. In view of (4.4) it is natural to require that

$$w \in \text{int}D \cap \text{int}C_\epsilon, \quad f(x) > f(w) \quad \forall x \in D \setminus \text{int}C_\epsilon \tag{4.9}$$

Define $\bar{\gamma} = \inf\{f(x) | x \in D, h(x) > -\epsilon\}$ and let $G = D(\bar{\gamma}),\ \Omega = D(\bar{\gamma}) \setminus \text{int}C_\epsilon$.

In view of (4.9) and Proposition 4.2, it is easily seen that $\Omega$ coincides with the set of $\epsilon$-approximate optimal solutions of CDC, so the problem amounts to searching for a point $\bar{x} \in \Omega$. Denote by $\mathcal{P}$ the family of polytopes $P \in \mathbb{R}^n$ for which there exists $\gamma \in [\bar{\gamma}, +\infty]$ satisfying $G \subset D(\gamma) \subset P$.

Consider the general problem of searching for an element of an unknown set $\Omega \subset \mathbb{R}^n$ (for instance, $\Omega$ is the set of optimal solutions of a given problem). Suppose there exist a closed set $G \supset \Omega$ and a family $\mathcal{P}$ of polyhedrons $P \supset G$, such that for each polyhedron $P \in \mathcal{P}$ a point $x(P) \in P$ (called *a distinguished point* associated with $P$) can be defined satisfying the following conditions:

1. $x(P)$ always exists and can be computed if $\Omega \neq \emptyset$, and whenever a sequence of distinguished points $x^1 = x(P_1),\ x^2 = x(P_2)$,..., converges to a point $\bar{x} \in G$ then $\bar{x} \in \Omega$ (in particular, $x(P) \in G$ implies that $x(P) \in \Omega$).
2. Given any distinguished point $x(P)$, $P \in \mathcal{P}$, we can recognize when $x(P) \in G$ and if $x(P) \notin G$, we can construct and affine function $l(x)$ (called a "cut") such that $P' = P \cap \{x | l(x) \leq 0\} \in \mathcal{P}$ and $l(x)$ strictly separates $x(P)$ from $G$, i.e. satisfies

$$l(x(P)) > 0, \quad l(x) \le 0 \, \forall x \in G. \tag{4.10}$$

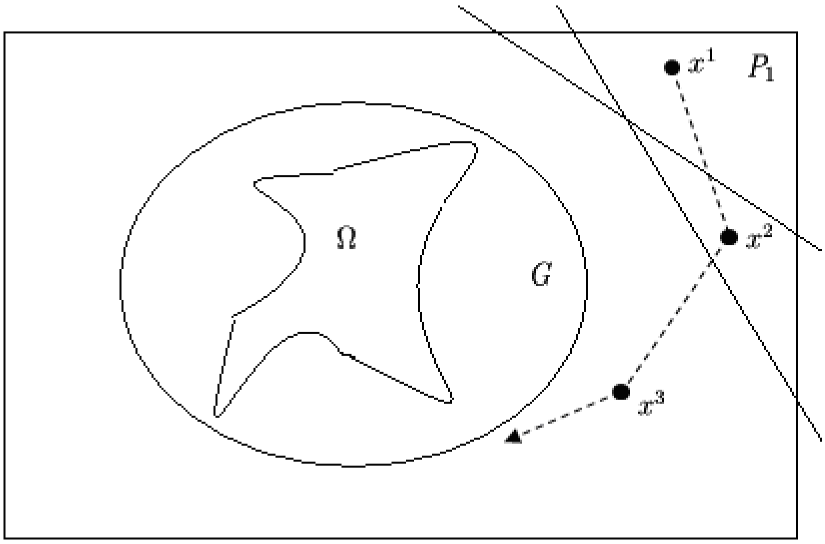


Figure 4.1 Outer approximation scheme for general nonconvex optimization.

Then, for every $P \in \mathcal{P}$, it is possible to define

$$x(P) \in \text{argmax}\{h(x)|x \in V\} \tag{4.11}$$

Where $V$ is the vertex set of $P$. We now verify the use of outer approximation scheme for problem CDC. First condition is obvious because $x(P)$ exists and can be computed provided $\Omega \neq \emptyset$; moreover, since $P \supset G \supset \Omega$, we must have $h(x(P)) \ge -\epsilon$, so any accumulation point $\bar{x}$ of a sequence $x(P_k),\ k = 1, 2, \ldots$, satisfies $h(\bar{x}) \ge -\epsilon$, and hence $\bar{x} \in \Omega$ whenever $\bar{x} \in G$. To verify second condition, let any $x(P)$ associated to a polytope $P$ such that $D(\gamma) \subset P$ for some $\gamma \in [\bar{\gamma}, +\infty]$. Note that $h(x(P)) \ge -\epsilon$. If $h(x(P)) < 0$ then $\max\{h(x)|x \in P\} < 0$, hence $D(\gamma) \subset \{x|h(x) < 0\}$, which implies that $\gamma$ is an $\epsilon$-approximate optimal value if $\gamma < +\infty$ or the problem is infeasible if $\gamma = +\infty$. If $h(x(P)) \ge 0$ then $\max\{g(x(P)),\ h(x(P)) + \epsilon\} > 0$ and since $\max\{g(w),\ h(w) + \epsilon\} < 0$, we can compute a point $y$ such that

$$y \in [w, x(P)], \quad \max\{g(y), h(y) + \epsilon\} = 0 \tag{4.12}$$

Two cases are possible:

a. $g(y)=0$: since $g(w)<0$ this event may occur only if $g(x(P))>0$ and so $x(P)$ can be separated from $G$ by a cut $l(x)=\mathbf{p}^T(x-y)\leq 0$ with $\mathbf{p}\in\partial g(x)/\partial x$ evaluated at $y$.

b. $h(y)=-\epsilon$: then $g(y)\leq 0$, so $y$ is an $\epsilon$-approximate solution in the sense of (4.7). Furthermore, since $y=(1-\lambda)w+\lambda x(P)$ with $0<\lambda<1$, it follows that $f(y)\leq(1-\lambda)f(w)+\lambda f(x(P))<f(x(P))$, so $x(P)$ can be separated from $G$ by a cut $l(x)=\mathbf{p}^T(x-y)\leq 0$ with $\mathbf{p}\in\partial g(x)/\partial x$ evaluated at $y$.

In either case, if we set $\gamma'=\min\{\gamma, f(y)\}$ then $P'=P\cap\{x|l(x)\leq 0\}\supset D(\gamma')$, i.e. $P'\in\mathcal{P}$. Thus an outer approximation scheme can be applied to solve CDC [22].

**Algorithm 4.1 OA algorithm for CDC.**

---

*Step* 0. Let $\bar{x}^1$ be the best feasible solution available, $\gamma_1=f(\bar{x}^1)$ (if no feasible solution is known, set $\bar{x}^1=\emptyset$, $\gamma_1=+\infty$). Take a polytope $P_0\supset D$ and let $P_1=\{x\in P_0|f(x)\leq\gamma_1\}$. Determine the vertex set $V_1$ of $P_1$. Set $k=1$.

*Step* 1. Compute $x^k\in$ argmax$\{h(x)|x\in V_k\}$. If $h(x^k)<0$, then terminate:

a) If $\gamma_k<+\infty$, $\bar{x}^k$ is an $\epsilon$-approximate optimal solution.

b) If $\gamma_k=+\infty$, the problem is infeasible.

*Step* 2. Compute $y^k\in[w,x^k]$ such that $\max\{g(y^k), h(y^k)+\epsilon\}=0$. If $g(y^k)=0$ then set $\gamma_{k+1}=\gamma_k$, $\bar{x}^{k+1}=\bar{x}^k$ and let

$$l_k(x):=\langle p^k, x-y^k\rangle,\quad p^k\in\partial g(y^k).$$

*Step* 3. If $h(y^k)=-\epsilon$ then set $\gamma_{k+1}=\min\{\gamma_k, f(y^k)\}$, $\bar{x}^{k+1}=y^k$ if $f(y^k)\leq\gamma_k$, $\bar{x}^{k+1}=\bar{x}^k$ otherwise. Let

$$l_k(x):=\langle p^k, x-y^k\rangle,\quad p^k\in\partial f(y^k).$$

*Step* 4. Compute the vertex set $V_{k+1}$ of $P_{k+1}=P_k\cap\{x|l_k(x)\leq 0\}$, set $k\leftarrow k+1$ and go back to *step* 1.

---

## 4.2 SIGNOMIAL GEOMETRIC PROGRAMMING

Signomial Geometric Programming (SGP) problem can be given as

$$\text{(SGP)} : \begin{cases} \text{Minimize} & F_0(x) \\ \text{subject to} & F_j(x) \leq 0, \quad j = 1, \ldots, M, \\ & x \in X = [\underline{x}, \overline{x}] \subset \mathbb{R}^N, \end{cases} \tag{4.13}$$

where

$$F_j(x) = \sum_{t=1}^{T_j} \delta_{jt} \alpha_{jt} \prod_{i=1}^{N} x_i^{\gamma_{jti}}, \quad j = 0, 1, \ldots, M, \tag{4.14}$$

and $\alpha_{jt}$ is a positive and real coefficient; $\delta_{jt}$ = +1 or -1; $\gamma_{jti}$ is an arbitrary real constant exponent; $\underline{x}$ and $\overline{x}$ are $N$-vectors with $\underline{x} > 0$. In general, SGP corresponds to a nonlinear optimization problem with nonconvex objective function and constraints.

SGP is a special nonlinear programming problem that has many applications in engineering design [23-26], economics and statistics [27-30], manufacturing [31,32] and chemical equilibrium [33,34]. There are many local optimization approaches for SGP, however the global optimization algorithms based on the characteristic of SGP are scarce. Maranas and Gloudas [33] proposed such a global optimization algorithm based on the exponential variable transformation of SGP, the convex relaxation and branching and bounding on some hyperrectangle region. By using linear relaxation, Shen and Zhang [35] reduce the problem SGP to a sequence of linear programming problems through successive refinement of a linear relaxation of feasible region using exponential variable transformation, tangential hypersurfaces and convex envelop approximations. They report efficient results for the global solution of SGP.

Another global optimization algorithm for SGP is proposed by Wang and Liang [36]. They use the popular exponential variable transformation to convert the problem into a

Reverse Convex Programming RCP problem. Then by successively approximating convex constraint with a linear constraint and using the linear relaxation of RCP, they propose a convergent cutting-plane algorithm and give robust results for famous SGP benchmark problems. However, Tuy [37] shows that the $\epsilon$-approximate solutions offered by the above relaxation schemes quite often tend to be far from the actual global optimum solution of SGP. Therefore, he proposes a DC programming and monotonic optimization procedure for a robust solution of generalized nonconvex optimization problems. Using the results due to Tuy, Shen *et al.* [38] provide another robust optimization algorithm for the solution of SGP. However, the performance of their algorithm varies on some of the problems encountered in literature.

Previous results in the global optimization of SGP using linear relaxation have been gathered by Shen *et al.* [21]. They propose an acceleration method using a suitable deletion technique. Their technique offers the possibility to cut away a significant portion of the currently investigated feasible region which does not contain the global minimum of SGP. Using the new deletion technique they report less number of iterations and the execution time of their algorithm is significantly reduced. We implement this algorithm for the solution of type I and type II problems consisting of MTTF and availability functions.

We apply the exponent variable transformation $x = \exp(y)$ and equivalently represent SGP as follows,

$$\text{(P1)}: \begin{cases} \text{Minimize} & \Psi_0(y) \\ \text{subject to} & \Psi_j(y) \leq 0, \quad j = 1, \ldots, M, \\ & y \in Y^0 = [\underline{y}^0, \overline{y}^0] \subset \mathbb{R}^N, \end{cases} \tag{4.15}$$

where

$$\Psi_j(y) = \sum_{t=1}^{T_j} \delta_{jt} \alpha_{jt} \exp(\sum_{i=1}^{N} \gamma_{jti} y_i), \quad j = 0, 1, \ldots, M, \tag{4.16}$$

and $\underline{y}^0 = ln\underline{x}$, $\overline{y}^0 = ln\overline{x}$.

### 4.2.1 Linear Relaxation

The principal construct in the development of a solution procedure for solving problem P1 is the construction of a linear relaxation for obtaining the lower bound of the optimal value for this problem, as well as for its partitioned subproblems. Such a linear relaxation can be realized by lower estimating every convex term and upper estimating ever concave term of each constraint, in either the initial bounds on the variables of the problem, or modified bounds as defined for some partitioned subproblems in a branch and bound scheme. In other words, this linear function is constructed by finding the linear lower bound function of each implicitly separable term $\delta_{jt}\alpha_{jt}\exp(\sum_{i=1}^{N}\gamma_{jti}y_i)$, $j = 1, \ldots, M; t = 1, \ldots, T_j$.

Let $Y_{jt} = \sum_{i=1}^{N}\gamma_{jti}y_i$, $Y_{jt}^l = \sum_{i=1}^{N}\min(\gamma_{jti}\underline{y}_i, \gamma_{jti}\overline{y}_i)$, $Y_{jt}^u = \sum_{i=1}^{N}\max(\gamma_{jti}\underline{y}_i, \gamma_{jti}\overline{y}_i)$ denote the lower and upper bound of $Y_{jt}$ in the hyper-rectangle $Y_i = [\underline{y}_i, \overline{y}_i]$ respectively. Consider a function $f_{jt}(y) = \exp(\sum_{i=1}^{N}\gamma_{jti}y_i) = \exp(Y_{jt})$ for any $y \in Y = [\underline{y}, \overline{y}] \subseteq Y^0$, where $j = 0, 1, \ldots, M$ and $t = 1, \ldots, T_j$. Then the following statements are valid:

i. Let $A_{jt} = \dfrac{exp(Y_{jt}^u) - exp(Y_{jt}^l)}{Y_{jt}^u - Y_{jt}^l}$. Then,

$$g_{jt}(y) = \exp(Y_{jt}^l) + A_{jt}(\sum_{i=1}^{N}\gamma_{jti}y_i - Y_{jt}^l) = \exp(Y_{jt}^l) + A_{jt}(Y_{jt} - Y_{jt}^l),$$
$$h_{jt}(y) = A_{jt}(1 + \sum_{i=1}^{N}\gamma_{jti}y_i - \ln A_{jt}) = A_{jt}(1 + Y_{jt} - \ln A_{jt}),$$

denote an affine concave envelope of $f_{jt}(y)$ and an affine function corresponding to a supporting hyperplane of the graph of $f_{jt}(y)$ over $Y$ parallel to $g_{jt}(y)$, respectively. In other words,

$$h_{jt}(y) \leq f_{jt}(y) \leq g_{jt}(y), \quad \forall y \in Y.$$

ii. The differences $\Delta^1_{jt}(y) = g_{jt}(y) - f_{jt}(y)$ and $\Delta^2_{jt}(y) = f_{jt}(y) - h_{jt}(y)$ satisfy $\max_{y \in Y} \Delta^1_{jt}(y) = \max_{y \in Y} \Delta^2_{jt}(y) = \exp(Y^l_{jt})(1 - z_{jt} + z_{jt} \ln z_{jt})$ where $\omega_{jt} = Y^u_{jt} - Y^l_{jt}$ and $z_{jt} = \dfrac{\exp(\omega_{jt}) - 1}{\omega_{jt}}$.

The details can be found in [35]. From this result, it follows that $g_{jt}(y)$ and $h_{jt}(y)$ converge to $f_{jt}(y)$ as $\omega_{jt} \to 0$. Now we can give the linear relaxation problem P2 related to P1 as follows:

$$(\text{P2}) : \begin{cases} \text{Minimize} & \Psi^R_0(y) \\ \text{subject to} & \Psi^R_j(y) \leq 0, \quad j = 1, \ldots, M, \\ & y \in Y^0 = [\underline{y}^0, \overline{y}^0] \subset \mathbb{R}^N, \end{cases} \tag{4.17}$$

where

$$\Psi^R_j(y) = \sum_{t=1}^{T_j} \alpha_{jt} \Psi^L_{jt}(y), \quad j = 0, 1, \ldots, M, \tag{4.18}$$

$$\Psi^L_{jt}(y) = \begin{cases} \delta_{jt} h_{jt}(y) & if \quad \delta_{jt} = 1, \\ \delta_{jt} g_{jt}(y) & if \quad \delta_{jt} = -1. \end{cases}$$

Based on the above linear under-estimators, every feasible point of P1 in sub-domain $Y$ is feasible for P2, and the objective function value of P2 is less than or equal to that of P1 for all points in $Y$. Thus, the minimum of P2 provides a valid lower bound for the globally optimal value of P1 over a partition set $Y$. Therefore we can use the linear relaxation problem P2 to derive a lower bound of the solution of P1, which can be calculated by solving P2 inside some rectangle defined by $Y = (Y_i)_{N \times 1} \subseteq Y^0$ with $Y_i = [\underline{y}_i, \overline{y}_i]$.

### 4.2.2 Deletion Technique

The accelerated deletion technique described in Shen *et al.* [21] is based upon on two important global optimality theorems in some hyper rectangle space $Y$. We now give brief descriptions of those theorems. To this end, let

$$\beta_i = \sum_{t=1}^{T_0} \delta_{0t}\alpha_{0t}\gamma_{0ti}A_{0t}, \quad i = 1, \ldots, N.$$

**Theorem 4.2**

*Assume that $\overline{\Psi}_0$ is a known upper bound of the optimal objective value $\Psi_0^*$ of P1, and let $Y = (Y_i)_{N\times 1}$ with $Y_i = [\underline{y}_i, \overline{y}_i]$ be a sub-rectangle of $Y^0$. If there exists some index $m \in \{1, 2, \ldots, N\}$ satisfying $\beta_m > 0$ and $\rho_m < \beta_m\overline{y}_m$, then there is no globally optimal solution of P1 over $Y^1$; if $\beta_m < 0$ and $\rho_m < \beta_m\underline{y}_m$ for some $m$ then there is no globally optimal solution of P1 over $Y^2$* [21], *where*

$$\rho_m = \overline{\Psi}_0 - \sum_{\substack{i=1\\ i\neq m}}^{N} \min\{\beta_i\underline{y}_i, \beta_i\overline{y}_i\} - \sum_{\substack{t=1\\ \delta_{0t}=1}}^{T_0} \delta_{0t}\alpha_{0t}A_{0t}(1-\ln A_{0t})$$

$$- \sum_{\substack{t=1\\ \delta_{0t}=-1}}^{T_0} \delta_{0t}\alpha_{0t}(\exp(Y_{0t}^l) - A_{0t}Y_{0t}^l), \quad m = 1, \ldots, N,$$

$$Y^1 = (Y_i^1)_{N\times 1} \subseteq Y \quad \text{with} \quad Y_i^1 = \begin{cases} Y_i & if \quad i \neq m \\ (\frac{\rho_m}{\beta_m}, \overline{y}_m] \cap Y_i & if \quad i = m, \end{cases}$$

$$Y^2 = (Y_i^2)_{N\times 1} \subseteq Y \quad \text{with} \quad Y_i^2 = \begin{cases} Y_i & if \quad i \neq m \\ [\underline{y}_m, \frac{\rho_m}{\beta_m}) \cap Y_i & if \quad i = m, \end{cases}$$

**Theorem 4.3**

*Assume that* $\underline{\Psi}_0$ *is a known upper bound of the optimal objective value* $\Psi_0^*$ *of P1, and let* $Y = (Y_i)_{N\times 1}$ *with* $Y_i = [\underline{y}_i, \overline{y}_i]$ *be a sub-rectangle of* $Y^0$*. If there exists some index* $m \in \{1, 2, \ldots, N\}$ *satisfying* $\beta_m > 0$ *and* $\tau_m > \beta_m \underline{y}_m$*, then there is no globally optimal solution of P1 over* $Y^3$*; if* $\beta_m < 0$ *and* $\tau_m > \beta_m \overline{y}_m$ *for some* $m$ *then there is no globally optimal solution of P1 over* $Y^4$ [21], *where*

$$\tau_m = \underline{\Psi}_0 - \sum_{\substack{i=1 \\ i\neq m}}^{N} \max\{\beta_i \underline{y}_i, \beta_i \overline{y}_i\} - \sum_{\substack{t=1 \\ \delta_{0t}=1}}^{T_0} \delta_{0t}\alpha_{0t}(\exp(Y_{0t}^l) - A_{0t}Y_{0t}^l)$$

$$- \sum_{\substack{t=1 \\ \delta_{0t}=-1}}^{T_0} \delta_{0t}\alpha_{0t}A_{0t}(1 - \ln A_{0t}), \quad m = 1, \ldots, N,$$

$$Y^3 = (Y_i^3)_{N\times 1} \subseteq Y \quad \text{with} \quad Y_i^3 = \begin{cases} Y_i & if \quad i \neq m \\ [\underline{y}_m, \frac{\tau_m}{\beta_m}) \cap Y_i & if \quad i = m, \end{cases}$$

$$Y^4 = (Y_i^4)_{N\times 1} \subseteq Y \quad \text{with} \quad Y_i^4 = \begin{cases} Y_i & if \quad i \neq m \\ (\frac{\tau_m}{\beta_m}, \overline{y}_m] \cap Y_i & if \quad i = m, \end{cases}$$

The proofs of Theorem 4.2 and Theorem 4.3 can be found in Shen *et al.* [21].

### 4.2.3 Branching

During each iteration of the algorithm, the branching process creates a more refined partition that cannot yet be excluded from further consideration in searching for a globally optimal solution of P1. In the branching process we are going to use simple bisection rule which is defined in Shen and Zhang [35]. This rule is sufficient to ensure convergence since it drives all the intervals shrinking to a singleton for all the variables. Bisection branching rule can be given as follows.

Consider any node sub-problem identified by the rectangle $Y' = [\underline{y}', \overline{y}'] \subseteq Y^0$. Let

$$p = \underset{i=1,\dots,N}{\text{argmax}}\{\overline{y}'_i - \underline{y}'_i\}.$$

We partition $Y'$ by bisecting the interval $[\underline{y}'_p, \overline{y}'_p]$ into the sub-intervals $[\underline{y}'_p, (\underline{y}'_p + \overline{y}'_p)/2]$ and $[(\underline{y}'_p + \overline{y}'_p)/2, \overline{y}'_p]$. With the help of above definitions we can formulate the global optimization algorithm proposed in Shen *et al.* [21]. Let $\text{LB}(Y_k)$ refer to the optimal objective function value of P2 for the sub-rectangles $Y^k$ and $y^k = y(Y^k)$ refer to an element of corresponding argmin.

**Algorithm 4.2 Modified branch and bound algorithm for SGP.**

---

*Step* 0. Given a convergence tolerance $\epsilon_c > 0$, a feasibility tolerance $\epsilon_f > 0$ and a deleting tolerance $\epsilon_d > 0$; iteration counter $k = 1$; the upper bound $\text{UB} = +\infty$; the active node set $Q_0 = \{Y^0\}$; the set of feasible points $F = \emptyset$. Solve P2 for $Y = Y^0$ to obtain the lower bound $\text{LB}_0 = \text{LB}(Y^0)$ and $y^0 = y(Y)$. If $y^0$ is feasible for P1, update $F$ and UB if necessary. If $\text{UB} - \text{LB}_0 \leq \epsilon_c$, stop, and $y^0$ is the globally optimal solution of P1. Otherwise, proceed to *Step* 1.

*Step* 1. If the midpoint $\hat{y}$ of $Y^k$ is feasible for P1, update $F$ and UB such that $F = F \cup \{\hat{y}\}$ and $\text{UB} = \min_{y \in F} \Psi_0(y)$; if $F \neq \emptyset$, the incumbent point is denoted by $b := \underset{y \in F}{\text{argmin}}\ \Psi_0(y)$;

*Step* 2. for $m = 1$ to $N$ do

*Step* 2.0. Calculate $\beta_m,\ \rho_m\ \tau_m$ as defined in Theorems 1 and 2 for $Y^k$;

if $\beta_m = 0$, then go to *Step* 3;

else if $\beta_m > 0$ then go to *Step* 2.1;

else if $\beta_m < 0$ then go to *Step* 2.3;

*Step* 2.1. if $\underline{y}_m \leq \frac{\rho_m}{\beta_m} < \overline{y}_m$, then

if $\overline{y}_m - \frac{\rho_m}{\beta_m} \leq \epsilon_d$, then go to *Step* 2.2;

else set $\overline{y}_m = \frac{\rho_m}{\beta_m}$, and go to *Step* 3;

if $\frac{\rho_m}{\beta_m} < \underline{y}_m$, then set $Q_k = Q_k \setminus Y^k$, and go to *Step* 6;

*Step* 2.2. if $\underline{y}_m < \frac{\tau_m}{\beta_m} \leq \overline{y}_m$, then

if $\frac{\tau_m}{\beta_m} - \underline{y}_m \leq \epsilon_d$, then

if $m < N$, then set $m = m + 1$, and go to *Step* 2.0;

else go to *Step* 3;

else set $\underline{y}_m = \frac{\tau_m}{\beta_m}$ and go to *Step* 3;

if $\frac{\tau_m}{\beta_m} > \overline{y}_m$, then set $Q_k = Q_k \setminus Y^k$, and go to *Step* 6.

*Step* 2.3. if $\underline{y}_m < \frac{\rho_m}{\beta_m} \leq \overline{y}_m$, then

if $\frac{\rho_m}{\beta_m} - \underline{y}_m \leq \epsilon_d$, then go to *Step* 2.4;

else set $\underline{y}_m = \frac{\rho_m}{\beta_m}$, and go to *Step* 3;

if $\overline{y}_m < \frac{\rho_m}{\beta_m}$, then set $Q_k = Q_k \setminus Y^k$, and go to *Step* 6;

*Step* 2.4. if $\underline{y}_m \leq \frac{\tau_m}{\beta_m} < \overline{y}_m$, then

if $\overline{y}_m - \frac{\tau_m}{\beta_m} \leq \epsilon_d$, then

if $m < N$, then set $m = m + 1$, and go to *Step* 2.0;

else go to *Step* 3;

else set $\overline{y}_m = \frac{\tau_m}{\beta_m}$ and go to *Step* 3;

if $\frac{\tau_m}{\beta_m} < \underline{y}_m$, then set $Q_k = Q_k \setminus Y^k$, and go to *Step* 6.

*Step* 3. According to the above rectangle bisection rule for $Y^k$, we can get two new sub-rectangles, and denote the set of new partition rectangles as $\overline{Y}^k$.

*Step* 4. For each $Y \in \overline{Y}^k$, compute the lower bound $\underline{\Psi}_j^R$ of $\Psi_j^R(y)$ over $Y$, i.e., $\underline{\Psi}_j^R = \sum_{t=1}^{T_j} \alpha_{jt} \underline{\Psi}_{jt}^L$, $j = 0, 1, \ldots, M$, where

$$\underline{\Psi}_{jt}^L = \begin{cases} \delta_{jt} h_{jt}(Y_{jt}^l) & if \quad \delta_{jt} = 1, \\ \delta_{jt} g_{jt}(Y_{jt}^u) & if \quad \delta_{jt} = -1. \end{cases}$$

If $\underline{\Psi}_0^R > \mathrm{UB}$ or $\underline{\Psi}_j^R > \epsilon_f$ for some $j \in \{1, \dots, M\}$, then the corresponding sub-rectangle $Y$ will be removed from $\overline{Y}^k$, i.e., let $\overline{Y}^k = \overline{Y}^k \setminus Y$ and skip to next element of $\overline{Y}^k$. If $\overline{Y}^k \neq \emptyset$, then solve P2 for each $Y \in \overline{Y}^k$ to obtain $\mathrm{LB}(Y)$ and $y(Y)$. If $\mathrm{LB}(Y) > \mathrm{UB}$ then $\overline{Y}^k = \overline{Y}^k \setminus Y$.

*Step* 5. If $y(Y)$ is feasible for P1, then update UB, $F$ and $b$ as *Step* 1. Set $Q_k = (Q_k \setminus Y^k) \cup \overline{Y}^k$ and the new lower bound $\mathrm{LB}_k = \min_{Y \in Q_k} \mathrm{LB}(Y)$.

*Step* 6. Set $Q_{k+1} = Q_k \setminus \{Y | \mathrm{UB} - \mathrm{LB(Y)} \leq \epsilon_c, Y \in Q_k\}$. If $Q_{k+1} = \emptyset$ then stop and UB is the optimal value of P1, $b$ is an optimal solution of P1. Otherwise, $k \leftarrow k+1$, select an active node $Y^k$ such that $Y^k \in \operatorname{argmin}_{Y \in Q_k} \mathrm{LB}(Y)$ and $y^k := y(Y^k)$ for further considering, and return to *Step* 1.

# 5 NUMERICAL EXAMPLES

In this section, we give some numerical examples to clarify the theoretical work. The main algorithm, branch and bound schemes are coded in C/C++ environment. The CDC algorithm is coded in MATLAB environment and implemented in C/C++ environment by using MATLAB C callable library generated by MATLAB compiler. The linear programming problems are solved using standard LP solver provided by CPLEX 11.1 callable library. The execution times are collected on a x64 HP workstation with 2.40 GHz dual CPU and 4096 MB RAM.

## 5.1 EXAMPLES FOR MTTF

In this section we provide two sets of numerical examples to illustrate the system-based component testing problems with MTTF performance measure. The first example set consists of four systems. Lower and upper bounds on component failure rates and unit component testing costs are given in Table 5.1. We set $\rho_M^1 = 3$, $\rho_M^2 = 10$, $\alpha = 0.05$ and $\beta = 0.05$ in all examples.

Table 5.1 First data set for MTTF examples.

| $j$ | redundant | | | $k$-out-of-$n$ | | | standby redundant | | | mixed | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ |
| 1 | 0.010 | 0.481 | 77.3 | 0.007 | 0.547 | 77.9 | 0.091 | 0.814 | 14.1 | 0.073 | 0.939 | 67.8 |
| 2 | 0.010 | 0.388 | 28.4 | 0.005 | 0.138 | 93.4 | 0.091 | 0.126 | 42.1 | 0.089 | 0.823 | 75.7 |
| 3 | 0.106 | 0.388 | 77.3 | 0.053 | 0.149 | 12.9 | 0.097 | 0.814 | 191 | 0.010 | 1.383 | 74.3 |

**Example 5.1**

The first example of this set is a serial connection of redundant subsystems with 2 components each. Component reliabilities and unit test costs are provided in Table 5.1.

We calculate the first feasible $m$ as 9 with $z_9^* = 4818.02$. Then $z_{10}^* = 5203.65$. Hence $m^* = 9$ and the corresponding test times are $(t_1^*, t_2^*, t_3^*) = (27.6762, 27.6705, 24.4674)$. The least reliable subsystem 1 is tested the most and between the similar subsystems 2 and 3, 3 is tested less as it has a higher testing cost. A total of 74 columns are generated in 490.70 CPU seconds. If we modify the unit component testing costs as $(c_1, c_2, c_3) = (7.3, 28.4, 147.3)$, the optimum $m$ becomes $m^* = 15$, with $z_{15}^* = 3785.36$. The optimum component test times are $(t_1^*, t_2^*, t_3^*) = (100.6966, 100.7719, 1.2645)$. The increased test time of subsystem 1 is a result of the reduction in subsystem 1's unit testing cost. However as a consequence of the longer test time of less reliable subsystem, the optimum number of observed component failures, namely $m^*$, is increased. This solution is illustrated Figure 5.1.

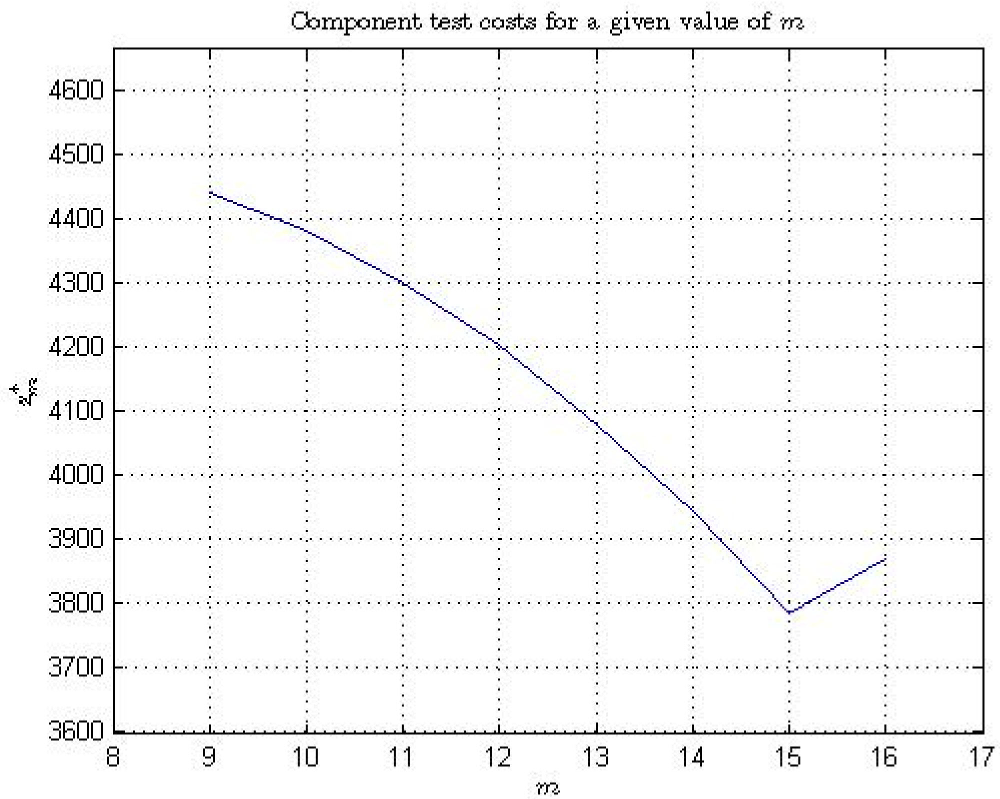


Figure 5.1 Sequence of total test costs versus $m$ for Example 5.1.

**Example 5.2**

As a second case, we study the serial connection of three 2-out-of-3 subsystems. Component reliabilities and unit test costs are provided in Table 5.1. The feasible $m$ is detected as 7 with $z_7^* = 6840.22$. Then $z_8^* = 6465.82$ and $z_9^* = 6754.59$. Therefore $m^* = 8$ and the corresponding test times are $(t_1^*, t_2^*, t_3^*) = (48.4039, 23.2431, 40.2906)$. A total of 70 columns are generated in 486.75 CPU seconds to find this solution. In this example, we increase the unacceptable expected system lifetime level from $\rho_M^1 = 3$ to $\rho_M^1 = 4$. As a result the optimum $m$ is 19 with total component testing cost $z_{19}^* = 17795.03$. The component test times are $(t_1^*, t_2^*, t_3^*) = (107.8319, 82.2468, 131.7244)$. The increase in total component testing cost and test times are notable, which is a clear result because we started to apply the probability of type I error to a wider range of unacceptable performance level. Similarly if we set $\rho_M^1 = 3$ and $\rho_M^2 = 9$, new optimum solution, $m^* = 10$, $z_{10}^* = 7615.77$, $(t_1^*, t_2^*, t_3^*) = (56.6176, 27.7525, 47.1262)$, follows the same result as well. Because we started to apply type II error probability, to a wider range of acceptable performance level. To clarify this result, let's recall type II problem given in (3.20). By decreasing the acceptable MTTF level $\rho_M^2$, we increase the size of the column generation set $\delta(\rho_M^2)$ given in (3.11). Therefore we are solving the $P(m)$ problem in a tighter, more constrained region, which results in a higher component testing cost.

**Example 5.3**

All subsystems of the third system are assumed to have a common redundancy $n_j = 2$. Component reliabilities and unit test costs are provided in Table 5.1. The column generation algorithm generates 37 columns in total to find $m^* = 7$ with $z_7^* = 3788.81$ and $(t_1^*, t_2^*, t_3^*) = (22.1353, 0.0000, 18.1379)$ in 6.79 CPU seconds. The problem has no feasible solution for smaller $m$ values. Arguments similar to the previous cases remain valid here. Subsystems 1 and 3 are quite similar and the one having higher unit test cost is less tested. The most reliable subsystem 2 is not tested at all. We modify the component failure rate lower bounds as $(lb_1, lb_2, lb_3) = (0.011, 0.101, 0.097)$. In the new setup, the column generation algorithm stops at $m = 7$ with increasing component testing cost $z_6^* = 3762.60$ and $z_7^* = 4177.45$, indicating $m^* = 6$. The component test times are $(t_1^*, t_2^*, t_3^*) = (16.4337, 0.0000, 18.4233)$. The shorter test time of subsystem 1 clearly depicts the increase in the reliability as a result of the change in lower bounds. However to balance the reduction in component testing time of subsystem 1, subsystem 3 is tested slightly longer as a tradeoff between two similar setups.

**Example 5.4**

In the fourth system, we investigate a mixed system that contains one redundant subsystem with 2 components, one standby redundant subsystem with 2 components and one 2-out-of-3 subsystem connected in series. Component reliabilities and unit test costs are provided in Table 5.1. The first feasible and optimum $m^* = 11$ and the total test cost is $z_{11}^* = 7915.06$. The component test times are $(t_1^*, t_2^*, t_3^*) = (32.1284, 23.8987, 52.7966)$. 72 columns are generated in 611.75 CPU seconds to solve this case.

In the second examples set, we investigate systems with subsystems having different redundancies. In line with the first set subsystems' redundancies, lower and upper bounds on component failure rates and unit component testing costs are given in Table 5.2. We set $\rho_M^1 = 3$, $\rho_M^2 = 10$, $\alpha = 0.05$ and $\beta = 0.05$ in all examples.

Table 5.2 Second data set for MTTF examples.

| $j$ | redundant | | | | $k$-out-of-$n$ | | | | standby redundant | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ |
| 1 | 4 | 0.014 | 0.043 | 70.6 | 2/3 | 0.010 | 0.092 | 32.9 | 4 | 0.059 | 0.236 | 48.9 |
| 2 | 3 | 0.042 | 0.168 | 3.18 | 2/4 | 0.013 | 0.312 | 32.9 | 3 | 0.059 | 0.845 | 44.5 |
| 3 | 2 | 0.091 | 0.649 | 27.6 | 3/4 | 0.029 | 0.312 | 87.4 | 2 | 0.164 | 0.845 | 64.6 |

**Example 5.5**

The first example of this set is a serial connection of redundant subsystems. Component reliabilities, unit test costs and subsystem redundancies are provided in Table 5.2. We calculate the first feasible $m$ with $z_7^* = 736.52$. Then $z_8^* = 808.58$. Therefore $m^* = 7$ and the corresponding test times are $(t_1^*, t_2^*, t_3^*) = (0.0000, 7.2068, 25.7681)$. Subsystem 1 with most reliable and more redundant components is not tested as it has longer expected lifetime. Component test times increase as subsystems reliabilities decrease. A total of 37 columns are generated in 3249.79 CPU seconds to find this solution.

**Example 5.6**

The second example consists of a serial connection of three $k$-out-of-$n$ subsystems. Component reliabilities, unit test costs and subsystem redundancies are provided in

Table 5.2. No feasible solution is detected $m \leq 9$. The optimum $m^* = 10$ with total test cost sequence $z^*_{10} = 9453.42$, $z^*_{11} = 10147.92$. The corresponding test times are $(t^*_1, t^*_2, t^*_3) = (26.4770, 38.9369, 83.4995)$. It can be again observed that the test times increase as reliabilities decrease. Comparing subsystems 1 and 2, the second requires less number of components to survive. However it is also less reliable according to the component reliabilities and thus needs more testing time. This solution is found in 1844.79 CPU seconds with the generation of 71 columns.

**Example 5.7**

The last example of this set is the serial connection of standby redundant subsystems. Component reliabilities, unit test costs and subsystem redundancies are provided in Table 5.2. We compute the first feasible $m$ as 9 with $z^*_9 = 2100.21$. Then we compute $z^*_{10} = 2268.31$. Therefore, $m^* = 9$ is the optimum solution of the component testing problem. The corresponding test times are $(t^*_1, t^*_2, t^*_3) = (0.0000, 12.8273, 23.6517)$. 44 columns were generated for this solution in 179.04 CPU seconds.

**Example 5.8**

In this example we compare passive redundancy with active redundancy. We design a system consisting three serially connected subsystems with 3 components each. We set $(lb_1, lb_2, lb_3) = (0.101, 0.107, 0.039)$, $(ub_1, ub_2, ub_3) = (0.765, 0.795, 0.186)$, $(c_1, c_2, c_3)$ $= (67.9, 65.5, 16.3)$, $\rho^1_M = 2$, and $\rho^2_M = 8$. First we obtain the solution for the passive redundant case, $m^* = 6$, $z^*_6 = 1979.69$ and $(t^*_1, t^*_2, t^*_3) = (14.0293, 14.1088, 6.2631)$. Then we solve the system for the active redundancy case, the solution is $m^* = 6$, $z^*_6 = 1335.32$ and $(t^*_1, t^*_2, t^*_3) = (9.3825, 9.6317, 4.0966)$. Here, smaller total component testing cost for the active redundancy case is an expected result, because standby redundant subsystems have a longer expected system lifetime.

**Example 5.9**

As previously mentioned, stopping at the first feasible $m$ for which the next feasible $m$ has higher total test cost turns out to be a good heuristic rule in practice. Though rarely occurs, it is also possible to find cases where this rule does not apply. As an example, let's consider a serial connection of three $k$-out-of-$n$ subsystems with common redundancies 2-out-of-3. Let $(lb_1, lb_2, lb_3) = (0.0056, 0.0266, 0.0173)$, $(ub_1, ub_2, ub_3) = (0.1818, 0.2638, 0.1455)$ and $(c_1, c_2, c_3) = (56.8823, 3.9390, 11.1902)$. Furthermore, unacceptable and acceptable expected system lifetimes are selected as $\rho_M^1 = 3$ and $\rho_M^2 = 12$ respectively. The first feasible $m = 7$ with $z_7^* = 2148.5424$. Given that the total test cost for the next $m = 8$ is $z_8^* = 2153, 4403$, the column generation algorithm stops with $m^* = 7$. However, if calculations are further carried, we find a sequence of total test costs such as 2148.54, 2153.44, 2157.36, 2155.88, 2145.51, 2126.19, 2098.58, 2063.98, 2022.79, 1975.31, 1955.68, 2046.94, 2137.81, 2228.33, 2318.51, 2408.39. Obviously, $m^* = 17$ with $z_{17}^* = 1955.68$. This case is illustrated in Figure 5.2.

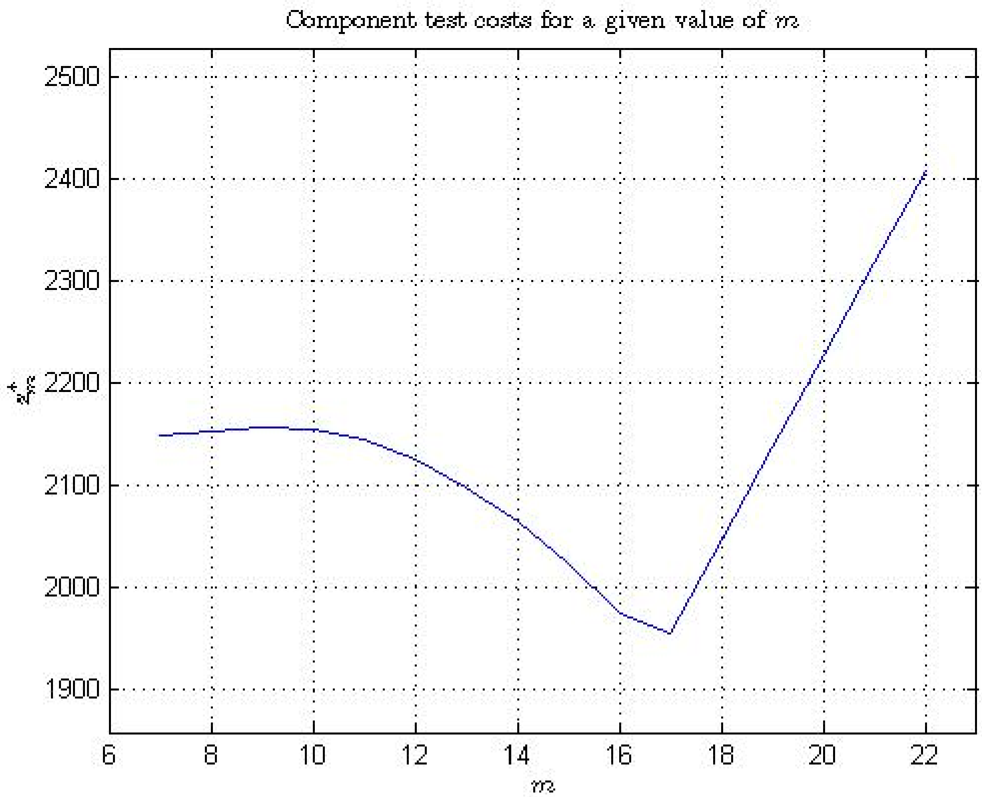


Figure 5.2 Sequence of total test costs versus $m$ for Example 5.9.

## 5.2 EXAMPLES FOR AVAILABILITY

In this section we formulate system-based component testing problem with availability performance measure. Again the first set of examples consists of systems having same redundancy in each subsystem. Lower and upper bounds on component failure rates and unit component testing costs are given in Table 5.3. We set $\rho_A^1 = 0.70$, $\rho_A^2 = 0.85$, $\alpha = 0.05$ and $\beta = 0.05$ in all examples.

Table 5.3 First data set for availability examples.

| $j$ | redundant | | | $k$-out-of-$n$ | | | standby redundant | | | mixed | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ |
| 1 | 0.032 | 0.187 | 44.5 | 0.037 | 0.247 | 77.9 | 0.083 | 0.421 | 82.3 | 0.173 | 0.830 | 34.9 |
| 2 | 0.032 | 0.795 | 44.5 | 0.075 | 0.138 | 33.4 | 0.083 | 0.728 | 38.9 | 0.104 | 0.585 | 19.6 |
| 3 | 0.095 | 0.187 | 64.6 | 0.073 | 0.149 | 42.9 | 0.093 | 0.728 | 74.3 | 0.180 | 0.549 | 25.1 |

**Example 5.10**

First of the availability examples is a system consisting of three serially connected redundant subsystems with $n_j = 2$ redundancy each. Component reliabilities and unit testing costs are provided in Table 5.3. We compute $m^* = 13$ with $z_{13}^* = 2247.88$ and the component test times corresponding to this solution as $(t_1^*, t_2^*, t_3^*) = (0.9964, 49.4514, 0.0000)$. 63 columns were generated in 127.53 CPU seconds to compute this solution. We can again observe the increase in test times with decreasing subsystem reliabilities. We change the unit test cost of subsystem 1 to $c_1 = c_3 = 64.6$, and we also set the upper bounds on component failure rates as $(lb_1, lb_2, lb_3) = (0.107, 0.795, 0.287)$. In the new solution optimum $m$ is found as $m^* = 17$, optimum total component testing cost $z_{17}^* = 3231.66$ and $(t_1^*, t_2^*, t_3^*) = (0.0000, 65.5852, 4.7853)$. An interesting result is that subsystem 1 is not tested as a result of its increased reliability. However, subsystem 3 is tested in the current solution because of its decreased reliability.

**Example 5.11**

Secondary, we inspect a serial connection of three 2-out-of-3 subsystems. Component reliabilities and unit testing costs are provided in Table 5.3. There is no feasible $m$ for $m \leq 6$. Then we compute $z_7^* = 4569.27$ and $z_8^* = 4873.92$ which indicates the optimum $m$ is $m^* = 7$. The optimum test times are $(t_1^*, t_2^*, t_3^*) = (55.1493, 8.1499, 0.0000)$. This solution is computed in 107.45 CPU seconds by generating 45 columns. In this solution it is again clear that the least reliable subsystem, namely subsystem 1, is the one with a longer test time. We set $\rho_A^1 = 0.69$, and the new optimum solution is, $m^* = 6$, $z_6^* = 3765.92$ and $(t_1^*, t_2^*, t_3^*) = (45.4213, 6.7912, 0.0000)$. The lesser total component testing cost is a clear result of the relaxation of type I problem. We started to apply the probability of type I error to a narrower range of unacceptable performance level. Similarly if we set $\rho_A^2 = 0.86$, the optimum solution is, $m^* = 4$, $z_4^* = 3213.68$ and $(t_1^*, t_2^*, t_3^*) = (39.6818, 3.6469, 0.0000)$. Again the lesser total component testing cost is a result of the relaxed type II problem.

**Example 5.12**

Third example involves a serial connection of 3 standby redundant subsystems, each with $n_j = 2$ redundant components. Component reliabilities and unit testing costs are provided in Table 5.3. The optimum $m^* = 18$ is computed with total test cost values $z_{18}^* = 4527.24$ and $z_{19}^* = 4701.97$. $(t_1^*, t_2^*, t_3^*) = (18.4728, 26.3364, 26.6628)$ are the optimum test times corresponding with this solution. 136 columns were generated in 468.98 CPU seconds to find this solution.

**Example 5.13**

The forth system consists the serial connection of one redundant subsystem with two components, one standby redundant subsystem with two components and one 2-out-of-3 subsystem. Component reliabilities and unit testing costs are provided in Table 5.3. The algorithm generates 103 columns in 950.10 CPU seconds to compute $z_{16}^* = 1374.93$ and $z_{17}^* = 1442.68$ which indicates the optimum $m$ is $m^* = 16$. The optimum test times are $(t_1^*, t_2^*, t_3^*) = (10.7204, 13.9076, 28.9273)$. In this example, we set $u_1 = u_3 = 0.663$. The optimum $m$ is calculated as $m^* = 16$ with total component testing cost sequence $z_{16}^* = 1699.61$, $z_{17}^* = 1769.46$. Also the component test times in this solution are $(t_1^*, t_2^*, t_3^*) = (8.7927, 12.7237, 32.6225)$. The increased reliability of subsystem 1 results in a shorter test time and the reduced reliability of subsystem 3 results in a longer test time in the new solution.

The second set consists subsystems with varying redundancy. The component reliabilities, unit test costs and subsystem redundancies are provided in Table 5.4.

Table 5.4 Second data set for availability examples.

| $j$ | redundant | | | | $k$-out-of-$n$ | | | | standby redundant | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ | $n_j$ | $lb_j$ | $ub_j$ | $c_j$ |
| 1 | 4 | 0.141 | 0.379 | 11.9 | 2/3 | 0.039 | 0.353 | 4.61 | 4 | 0.163 | 0.278 | 96.4 |
| 2 | 3 | 0.201 | 0.615 | 49.8 | 2/4 | 0.065 | 0.116 | 9.71 | 3 | 0.181 | 0.546 | 15.7 |
| 3 | 2 | 0.195 | 0.362 | 95.9 | 3/4 | 0.017 | 0.138 | 82.3 | 2 | 0.025 | 0.957 | 97.0 |

**Example 5.14**

The first example of second data set is a system consisting of three serially connected redundant subsystems with component reliabilities, unit testing costs and subsystem redundancies provided in Table 5.4. We compute $m^* = 13$ with $z^*_{13} = 1928.59$ and the component test times corresponding to this solution as $(t^*_1, t^*_2, t^*_3) = (1.1725, 38.4185, 0.0000)$. 57 columns were generated in 3012.50 CPU seconds to compute this solution. We can again observe the increase in test times with decreasing subsystem reliabilities.

**Example 5.15**

Secondly we investigate a serial connection of three $k$-out-of-$n$ subsystems. Component reliabilities, unit testing costs and subsystem redundancies are provided in Table 5.4. The column generation algorithm computes the total test cost sequence illustrated in Figure 5.3, starting from $m = 14$, which indicates $m^* = 33$ with $z^*_{33} = 1003.88$. The component test times corresponding to this solution are $(t^*_1, t^*_2, t^*_3) = (217.4265, 0.0000, 0.0000)$. 145 columns were generated in 6578.31 CPU seconds to compute this solution.

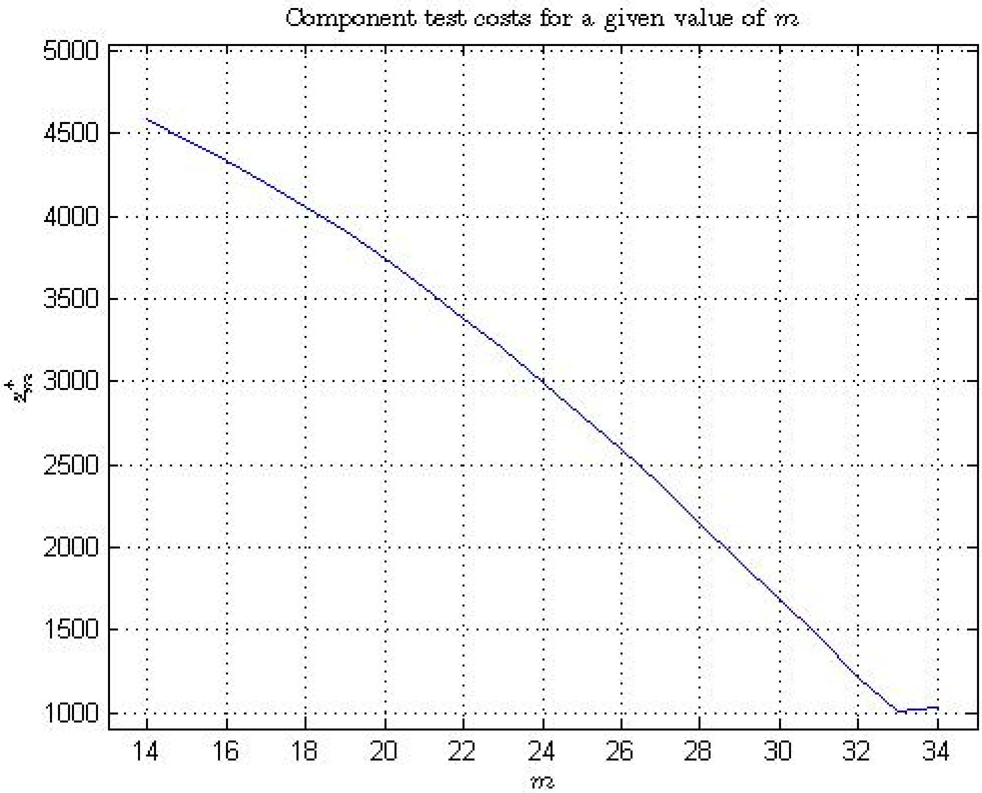


Figure 5.3 Sequence of total test costs versus $m$ for Example 5.15.

**Example 5.16**

The final example of second set is a serial connection of 3 standby redundant subsystems. Component reliabilities, unit testing costs and subsystem redundancies are provided in Table 5.4. There is no feasible $m$ for $m \leq 13$, then we compute $z_{14}^{*} = 2480.35$ and $z_{15}^{*} = 2607.87$. Hence the column generation algorithm stops with $m^{*} = 14$ and $(t_1^*, t_2^*, t_3^*) = (0.0000, 9.8657, 23.9529)$. 99 columns were generated in 648.46 CPU seconds to find this solution. We observe the most reliable system subsystem 1 is not tested and the comparison between subsystem 2 and 3 is clear as the less reliable one, namely 3 is tested longer.

## 5.3 EXAMPLES FOR SMPM

In this section, we formulate system-based component testing problems taking all three performance measures into account, namely we set $Z = \{R, M, A\}$. Let $\rho_l^1$ and $\rho_l^2$ for

$l \in Z = \{R, M, A\}$ denote the unacceptable and acceptable performance levels respectively.

Table 5.5 Data set for SMPM examples.

| $j$ | redundant | | | $k$-out-of-$n$ | | | standby redundant | | | mixed | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ | $lb_j$ | $ub_j$ | $c_j$ |
| 1 | 0.011 | 0.334 | 34.0 | 0.005 | 0.067 | 86.9 | 0.075 | 0.699 | 54.7 | 0.081 | 1.188 | 95.7 |
| 2 | 0.049 | 0.438 | 58.5 | 0.023 | 0.209 | 86.9 | 0.050 | 0.699 | 13.8 | 0.090 | 1.439 | 48.5 |
| 3 | 0.095 | 0.438 | 34.0 | 0.048 | 0.209 | 57.9 | 0.050 | 0.959 | 14.9 | 0.012 | 0.983 | 80.0 |

**Example 5.17**

In the first example we analyze the serial connection of three redundant subsystems with component reliabilities given in Table 5.5. We set $\rho_R^1 = 0.75$, $\rho_M^1 = 2$, $\rho_A^1 = 0.65$, and $\rho_R^2 = 0.95$, $\rho_M^2 = 12$, $\rho_A^2 = 0.90$ as unacceptable and acceptable performance measures, respectively. The redundancies of all subsystems are set to a common redundancy, i.e. $n_j = 2$ for all $j = 1, 2, 3$. No feasible solution is found for $m \leq 15$. Then, we compute $z_{16}^* = 3055.03$ and $z_{17}^* = 3208.40$ and deduce that $m^* = 16$. The associated optimum test times are $(t_1^*, t_2^*, t_3^*) = (11.5532, 24.9227, 35.3460)$. A total of 277 columns are generated in 971.05 CPU seconds to find this solution. Note that unit test costs are equal but lower and upper bounds are different for subsystem 1 and 3 and more reliable subsystem 1 has a shorter test time. Meanwhile, subsystem 2 and 3 are quite similar in terms of lower and upper bounds of component failure rates, but subsystem 2 with higher unit test cost is tested less.

We modify the desired performance levels as follows, $\rho_R^1 = 0.75$, $\rho_M^1 = 1.999$, $\rho_A^1 = 0.649$, $\rho_R^2 = 0.951$, $\rho_M^2 = 12$ and $\rho_A^2 = 0.901$, and obtain the solution, $m^* = 16$, $z_6^* = 3058.10$ and $(t_1^*, t_2^*, t_3^*) = (11.3755, 26.0314, 33.7076)$. From this point on, we add the additional modification of acceptable MTTF level, $\rho_M^2 = 12.01$ and an interesting result occurs. The solution remains precisely the same, which means that type II problem including the MTTF constraint is redundant with this data set. Therefore we can remove it from the column generation scheme without changing the solution.

**Example 5.18**

Our second example consists of a serial connection of three 2-out-of-3 systems. Let the component failure rates and unit testing costs for each subsystem be as in Table 5.5. We set $\rho_R^1 = 0.85$, $\rho_M^1 = 3$, $\rho_A^1 = 0.72$, $\rho_R^2 = 0.98$, $\rho_M^2 = 13$, and $\rho_A^2 = 0.95$. The first feasible $m$ is detected as 12 with $z_{12}^* = 9721.78$. Then $z_{13}^* = 9631.94$ and $z_{14}^* = 10199.51.78$. Therefore $m^* = 13$ and the corresponding test times are $(t_1^*, t_2^*, t_3^*) = (0.0000, 66.4787, 66.4647)$. A total of 216 columns are generated in 2758.31 CPU seconds to find this solution. We can note that among similar systems subsystem 1 and subsystem 2, the less reliable one, namely the second subsystem is tested more. If we modify the component test costs as $(c_1, c_2, c_3) = (6.9, 186.9, 57.9)$, the optimum solution changes to, $m^* = 12$, $z_{12}^* = 13858.57$ and $(t_1^*, t_2^*, t_3^*) = (28.8914, 54.4782, 59.9409)$. In this solution we can note that the increased test time of the reliable subsystem, namely subsystem 1, resulted in a lower optimum $m$.

**Example 5.19**

The third system consists of three standby redundant subsystems with $n_j = 2$ each, and the component failure rates are given in Table 5.5. For the third example the performance levels are set as $\rho_R^1 = 0.55$, $\rho_M^1 = 2$, $\rho_A^1 = 0.60$, $\rho_R^2 = 0.95$, $\rho_M^2 = 15$, and $\rho_A^2 = 0.90$. The column generation algorithm generates 305 columns in total to find $m^* = 20$ with $z_{20}^* = 1848.97$ and $(t_1^*, t_2^*, t_3^*) = (20.5276, 24.7465, 25.6290)$ in 1054.46 CPU seconds. The problem has no feasible solution for smaller $m$ values. Arguments similar to the previous case remain valid here. Among subsystems 2 and 3, the more reliable system 2 is tested less and among the similar subsystems 1 and 2, the one with the larger test cost, subsystem 1, is tested less.

When we modify the upper bounds for this example as $(ub_1, ub_2, ub_3) = (1.099, 0.299, 0.959)$, the optimum solution changes to $m^* = 20$, $z_{12}^* = 2058.25$ and $(t_1^*, t_2^*, t_3^*) = (27.3597, 11.7035, 26.7158)$. The increased component failure rate for subsystem 1 results in a longer test time, similarly also the decreased component failure rate for subsystem 2 results in a shorter test time.

**Example 5.20**

Finally we investigate a mixed system that contains one redundant subsystem with two components, one standby redundant subsystem with two components and one 2-out-of-3 subsystem connected in series. Again the component reliabilities and unit test costs for each subsystem are set to the values provided in Table 5.5. Let the performance levels for fourth system be $\rho_R^1 = 0.45$, $\rho_M^1 = 1$, $\rho_A^1 = 0.50$, $\rho_R^2 = 0.95$, $\rho_M^2 = 10$, and $\rho_A^2 = 0.90$. The first feasible and optimum $m^* = 19$ and the total test cost is $z_{19}^* = 7221.06$. The component test times are $(t_1^*, t_2^*, t_3^*) = (7.9170, 28.5287, 63.4968)$. 290 columns are generated in 4458.02 CPU seconds to solve this case. Upon changing the desired performance levels to $\rho_R^1 = 0.20$, $\rho_M^1 = 0.80$, $\rho_A^1 = 0.495$, $\rho_R^2 = 0.98$, $\rho_M^2 = 11$, and $\rho_A^2 = 0.92$, hence relaxing the optimization problem, we calculate the optimum solution as $m^* = 3$, $z_3^* = 10050.58$ and $(t_1^*, t_2^*, t_3^*) = (1.9058, 3.1000, 8.4039)$. The decreased total component testing cost is a clear result of relaxed performance levels.

## 5.4 EXAMPLES FOR JMPM

**Example 5.21**

We now illustrate this case for a series system with two components. The reliability, MTTF and availability functions are

$$R(\lambda) = e^{-(\lambda_1+\lambda_2)t},\ E[L] = \frac{1}{\lambda_1+\lambda_2},\ \text{and } A = \frac{1}{1+\lambda_1+\lambda_2}, \tag{5.1}$$

respectively. Using the above performance functions we can derive the semi-infinite linear programming problem $P'(m)$ as follows,

$P'(m)$ :

$$\begin{aligned} \min \quad & c_1 t_{1,m} + c_2 t_{2,m}, \\ \text{s.t.} \quad & \end{aligned} \tag{5.2}$$

$$\left\{ \begin{array}{ll} \min & \lambda_1 t_{1,m} + \lambda_2 t_{2,m} \\ \text{s.t.} & \lambda_1 + \lambda_2 \geq -\ln \rho_R^1 \\ & \rho_M^1(\lambda_1 + \lambda_2) \geq 1 \\ & \rho_A^1(\lambda_1 + \lambda_2) \geq 1 - \rho_A^1 \\ & lb_1 \leq \lambda_1 \leq ub_1 \\ & lb_1 \leq \lambda_1 \leq ub_1 \end{array} \right\} \geq \lambda_{\alpha,m}, \tag{5.3}$$

$$\left\{ \begin{array}{ll} \max & \lambda_1 t_{1,m} + \lambda_2 t_{2,m} \\ \text{s.t.} & \lambda_1 + \lambda_2 \leq \ln \rho_R^2 \\ & \rho_M^2(\lambda_1 + \lambda_2) \leq 1 \\ & \rho_A^2(\lambda_1 + \lambda_2) \leq 1 - \rho_A^2 \\ & lb_1 \leq \lambda_1 \leq ub_1 \\ & lb_1 \leq \lambda_1 \leq ub_1 \end{array} \right\} \leq \lambda_{1-\beta,m}, \tag{5.4}$$

$$t_{1,m} \geq 0 \quad t_{2,m} \geq 0. \tag{5.5}$$

In the above formulation, type I and type II problems are both linear programming problems. For this example we set $(lb_1, lb_2) = (0.039, 0.013)$, $(ub_1, ub_2) = (1.981, 0.519)$, $(c_1, c_2) = (34.03, 58.52)$, and the desired performance levels as $\rho_R^1 = 0.65$, $\rho_M^1 = 3$, $\rho_A^1 = 0.60$, $\rho_R^2 = 0.90$, $\rho_M^2 = 10$, and $\rho_A^2 = 0.85$.

The optimum solution $m^* = 3$, $z_3^* = 1076.40$ and $(t_1^*, t_2^*) = (11.6304, 11.6304)$ for the JMPM case (Figure 5.4) is found in 0.26 CPU seconds with the generation of 17 columns.

For the SMPM case (Figure 5.5), we find the optimum solution $m^* = 27$, $z_{27}^* = 10338.06$ and $(t_1^*, t_2^*) = (111.7024, 111.7024)$ in 1.59 CPU seconds by generating 273 columns.

We now illustrate the generated columns for each level of $m$ and compare the results of two formulations.

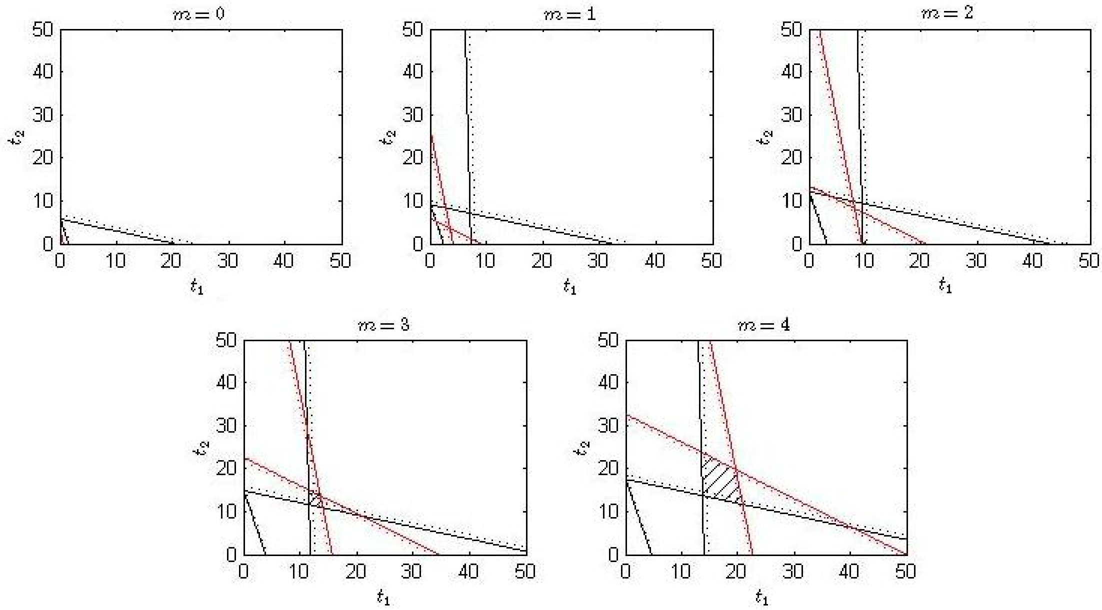


Figure 5.4 Generated columns for the joint case JMPM.

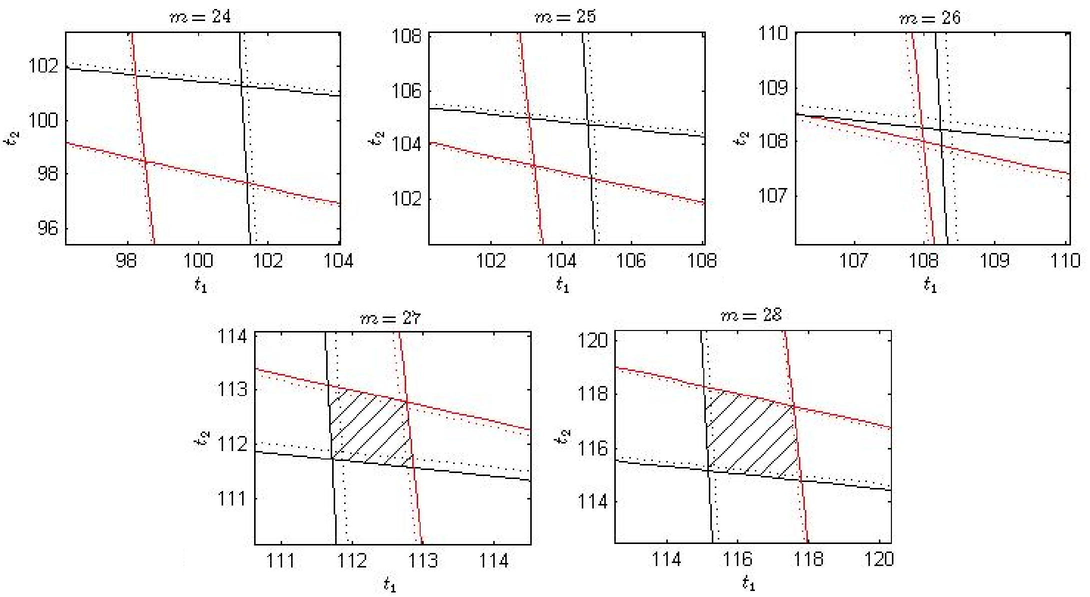


Figure 5.5 Generated columns for the separate case SMPM.

As we include more constraints with the same confidence interval in the SMPM formulation, it is also a tighter problem. Therefore the smaller total component testing cost found by the JMPM formulation is a result of its relaxed formulation compared to its SMPM counterpart.

## 6 CONCLUSION

The determination of cost efficient test plans to accept or reject a system with minimum total test cost is an important concern in reliability testing. These plans become more critical when it is impossible or economically infeasible to test the system as a whole. In the existing literature on system-based component testing, system reliability is regarded as a single performance measure. In this work, we generalize this concept to a multi-performance measure environment, where system reliability, expected system lifetime and system availability need to be demonstrated to accept or reject a system. We formulate expected system lifetime and system availability for various system topologies including the serial connection of redundant, $k$-out-of-$n$ and active redundant subsystems.

Type I and type II subproblems arising in the main formulation are reverse convex optimization problems and can be solved to global optimality using outer approximation exploiting this structure. However expected system lifetime and availability functions are rather complicated for this case and it is a well-known fact that outer approximation grows cumbersome with the increase in dimension. Therefore, we prefer to work with another representation involving signomial geometric programming and we choose a branch and bound algorithm with a relaxation scheme to solve these optimization problems.

We have obtained some notable results, such as

- the component test times increase as subsystem reliabilities decrease,
- a counter example to the stopping criterion for the column generation algorithm is also provided. It is also shown that stopping at the first feasible $m$ with increasing total component testing cost may not be appropriate for some cases,
- the subsystem requiring more components to survive is tested more,

- among subsystems with similar component reliabilities, the one with the smaller unit testing cost is tested more.

Furthermore, probably the most important result, the JMPM formulation gives a better solution with smaller total test cost and smaller first feasible $m$, compared to SMPM formulation. Therefore, an interesting line of research lies in formulating JMPM problem for different system topologies and comparing the results.

Another future line of research can be to deal with the variation of the time to failure for various systems. It is a common fact that a system with smaller expected system lifetime and time to failure variance is preferred in practice. If the variance can be explicitly expressed in functional form and bounded, it can be considered as an additional constraint in type I and type II problems involving expected system lifetime function.

It has been proven in the literature that, setting upper bounds on component failure rates is beneficial to reduce total test cost. However, it is assumed that upper bounds are obtained without any additional cost. The case where upper bounds are obtained at a price is an interesting line of research that is not addressed.

Finally, we want to point that the increasing complexity of expected system lifetime and availability functions also increases the intractability of the problem. In spite of this fact, we certainly hope that the analysis of this new type of testing problems leads to interesting stochastic models and optimization results. The new testing approach provides a more realistic setting to determine optimal component testing policies for devices that are designed to satisfy a set of performance measures.

# REFERENCES


[1] Barlow R. E., Proschan F., *Statistical Theory of Reliability and Life Testing*, Holt, Rinehart and Winston, New York, (1975).

[2] Barlow R., Proschan F., “Theory of maintained systems - Distribution of time to first system failure”, *Mathematics of Operations Research*, 1, 33-42, (1976).

[3] Çekyay B., Özekici S., “Reliability, MTTF and availability analysis of systems with exponential lifetimes”, *Koç University Technical Report*, (2008).

[4] Angus J., “On computing MTBF for k-out-of-n: G repairable system”, *IEEE Transactions on Reliability*, 37 (3), 312-313, (1988).

[5] Natarajan R., “A reliability problem with spares and multiple repair facilities”, *Operations Research*, 16 (5), 1041-1057, (1968).

[6] Gal S., “Optimal test design for reliability demonstration”, *Operations Research* 22, 1236-1242, (1974).

[7] Mazumdar M., “An optimum procedure for component testing in the demonstration of series system reliability”, *IEEE Transactions on Reliability*, R-26, 342-345, (1977).

[8] Easterling R. G., Mazumdar M., Spencer F. W., Diegert K. V., “System based component test plans for operating characteristics: binomial data”, *Technometrics*, 33 (3), 287-299, (1991).

[9] Altınel İ. K., “The design of optimum component test plans in demonstration of a series system reliability”, *Computational Statistics and Data Analysis*, 14, 281-292, (1992).

[10] Altınel İ. K., Özekici S., “A dynamic model for component testing”, *Naval Research Logistics*, 44, 187-197, (1997).

[11] Altınel İ. K., Özekici S., “Optimum component test plans for systems with dependent components”, *European Journal of Operational Research*, 111, 175-186, (1998).

[12] Çınlar E., Özekici S., "Reliability of a complex device in random environment", *Probability in the Engineering and Informational Sciences*, 1, 97-115, (1987).

[13] Altınel İ. K., Özekici S., Feyzioğlu O., "Component testing of repairable systems in multistage missions", *Journal of Operations Research Society*, 52, 937-944, (2001).

[14] Altınel İ. K., Özekici S., Feyzioğlu O., "Component testing of a series system in a random mission", *Reliability Engineering and System Safety*, 78 (1), 33-43, (2002).

[15] Feyzioğlu O., Altınel İ. K., Özekici S., "The design of optimum component test plans for system reliability", *Computational Statistics and Data Analysis*, 50, 3099-3112, (2006).

[16] Feyzioğlu O., Altınel İ. K., Özekici S., "Optimum component test plans for phased-mission systems". *European Journal of Operations Research*, 185, 255-265, (2008).

[17] Altınel İ. K., *System based component test problem: The design of optimum system based component test plans*, Ph.D. Thesis, Department of Industrial Engineering, University of Pittsburgh, (1990).

[18] Altınel İ. K., "The design of optimum component test plans in demonstration of system reliability", *European Journal of Operations Research*, 78, 318-333, (1994).

[19] Goberna M. A., Lopez M. A., *Linear Semi-infinite Optimization*, Wiley, Chichester, (1998).

[20] Horst R., Tuy H., *Global optimization: deterministic approaches*, Springer-Verlag, Berlin, (1996).

[21] Shen P., Li X., Jiao H., "Accelerating method of global optimization for signomial geometric programming", *Journal of Computational and Applied Mathematics*, 214, 66-77, (2008).

[22] Tuy H., *Convex analysis and global optimization*, Kluwer Academic Publishers, Dordrecht, (1998).

[23] Avriel, M., Williams, A.C., "An extension of geometric programming with applications in engineering optimization", *Journal of Engineering Mathematics*, 5(3), 187–199, (1971).

[24] Jefferson, T.R., Scott, C.H., "Generalized geometric programming applied to problems of optimal control: I. Theory", *Journal of Optimization Theory and Applications*, 26, 117–129, (1978).

[25] Nand, K.J., "Geometric programming based robot control design", *Computers and Industrial Engineers*, 29(1–4), 631– 635, (1995).

[26] Das, K., Roy, T.K., Maiti, M., "Multi-item inventory model with under imprecise objective and restrictions: a geometric programming approach", *Production Planning and Control*, 11(8), 781–788, (2000).

[27] Jae Chul, C., Bricker Dennis, L., "Effectiveness of a geometric programming algorithm for optimization of machining economics models", *Computers and Operations Research*, 23(10), 957–961, (1996).

[28] EI Barmi, H., Dykstra, R.L., "Restricted multinomial maximum likelihood estimation based upon Fenchel duality", *Statistics and Probability Letters*, 21, 121–130, (1994).

[29] Bricker, D.L., Kortanek, K.O., Xu, L., "Maximum Likelihood Estimates with Order Restrictions on Probabilities and Odds Ratios: A Geometric Programming Approach", *Applied Mathematical and Computational Sciences*, the University of IA, Iowa City, IA, (1995).

[30] Jagannathan, R., "A stochastic geometric programming problem with multiplicative recourse", *Operations Research Letters*, 9, 99–104, (1990).

[31] Sönmez, A.I., Baykasoglu, A., Dereli, T., Filiz, I.H., "Dynamic optimization of multipass milling operations via geometric programming", *International Journal of Machine Tools and Manufacture*, 39, 297–320, (1999).

[32] Scott, C.H., Jefferson, T.R., "Allocation of resources in project management", *International Journal of Systems Science*, 26, 413–420, (1995).

[33] Maranas C. D., Floudas C. A., "Global optimization in generalized geometric programming", *Compututers and Chemical Engineers*, 21 (4), 351–369, (1997).

[34] Rijckaert, M.J., Martens, X.M., "Analysis and optimization of the Williams-Otto process by geometric programming", *AIChE Journal*, 20(4), 742–750, (1974).

[35] Shen P. P., Zhang K. C., "Global optimization of signomial geometric programming using linear relaxation", *Applied Mathematics and Compututation*, 150, 99–114, (2004).

[36] Wang Y., Liang Z., "A deterministic global optimization algorithm for generalized geometric programming", *Applied Mathematics and Computation*, 168, 722-737, (2005).

[37] Tuy H., "Robust solution of nonconvex global optimization problems", *Journal of Global Optimization*, 32, 357–374, (2005).

[38] Shen P., Ma Y., Chen Y., "A robust algorithm for generalized geometric programming", *Journal of Global Optimization*, 41, 593-612, (2008).

# BIOGRAPHICAL SKETCH

Emre YAMANGİL is a Master of Science candidate in Industrial Engineering at Galatasaray University. He was born on December $17^{th}$, 1984, in İzmir / Turkey. He graduated from İzmir Anatolian Highschool and he holds a B.S. degree in Industrial Engineering from Dokuz Eylül University. He has presented a conference paper in YAEM’08 and he has been accepted to present a paper in YAEM’09. He has another conference paper among the proceedings of IESM’09. His current research involves applications of global optimization methods for non-convex optimization and he is a research fellow supported by Turkish Scientific and Technological Research Council’s 106M044 Grant.