

**ZAMANLANDIRILMIŐ PETRİ AĐLARINDA  
SİSTEM IKMAZI ANALİZİ**

**Zafer Hüseyn ERGAN**  
Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü  
Elektrik-Elektronik MühendisliĐi Anabilim Dalı  
AĐustos – 2004

## JÜRİ VE ENSTİTÜ ONAYI

Zafer Hüseyin ERGAN'ın Zamanlandırılmış Petri Ağlarında Sistem Çıkmazı Analizi başlıklı Elektrik-Elektronik Mühendisliği Anabilim Dalındaki, Yüksek Lisans tezi 05.08.2004 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı)	: Yard. Doç. Dr. Aydın AYBAR	
Üye	: Prof. Dr. Altuğ İFTAR	
Üye	: Yard. Doç. Dr. Cüneyt AKINLAR	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 08.09.2004... tarih ve ...29/7..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Altuğ İFTAR  
Enstitü Müdürü

**ÖZET****Yüksek Lisans Tezi****ZAMANLANDIRILMIŞ PETRİ AĞLARINDA SİSTEM ÇIKMAZI  
ANALİZİ****ZAFER HÜSEYİN ERGAN****Anadolu Üniversitesi****Fen Bilimleri Enstitüsü****Elektrik-Elektronik Mühendisliği Anabilim Dalı****Danışman: Yard. Doç. Dr. Aydın AYBAR****2004, 93 sayfa**

Bu tezde, kesikli olay sistemlerini modelleme yöntemlerinden biri olan zamanlandırılmış Petri ağları ele alınmıştır. Bu modelde, aynı anda veya devam etmekte olan ateşlemeler mevcut iken birden fazla geçiş ateşlenebilmektedir. Zamanlandırılmış Petri ağlarında, belirli zaman aralığı için ulaşılabilirlik kümesinin nasıl elde edileceği verilmiştir. Ayrıca zamanlandırılmış Petri ağları için sistem çıkmazı kavramına değinilmiş ve sistem çıkmazının nasıl tespit edileceği açıklanmıştır. Ulaşılabilirlik kümesinin oluşturulması ve sistem çıkmazının tespit edilmesi ile ilgili algoritmalar geliştirilmiş ve bu algoritmalar MATLAB programı kullanılarak gerçekleştirilmiştir.

**Anahtar Kelimeler: Kesikli Olay Sistemleri, Zamanlandırılmış Petri Ağları,  
Ulaşılabilirlik Kümesi, Sistem Çıkmazı.**

**ABSTRACT****Master of Science Thesis****DEADLOCK ANALYSIS FOR TIMED PETRI NETS****ZAFER HÜSEYİN ERGAN****Anadolu University****Graduate School of Natural and Applied Sciences  
Electrical and Electronics Engineering Program****Supervisor: Asst. Prof. Aydın AYBAR****2004, 93 pages**

**In this thesis, timed Petri nets which is one of the modeling methods of discrete event systems, are considered. In this model, it is assumed that, two or more transitions may fire at the same time or while existing firings are continuing. The construction of the reachability set for a given time interval in timed Petri nets is given. Moreover, deadlock concept for timed Petri nets is mentioned and deadlock detection is explained. Algorithms for reachability set construction and deadlock detection are developed and these algorithms are realized using MATLAB program.**

**Keywords: Discrete Event Systems, Timed Petri Nets, Reachability Set, Deadlock.**

## TEŐEKKÜR

Çalıőmalarım sırasında bana yol gösteren, bilgi ve deneyimiyle yardımlarını esirgemeyen tez danışmanım sayın Yrd. Doç. Dr. Aydın AYBAR'a ve beni her zaman destekleyen aileme sonsuz teőekkürlerimi sunarım.

**İÇİNDEKİLER**

	<u>Sayfa</u>
<b>ÖZET.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>TEŞEKKÜR.....</b>	<b>iii</b>
<b>İÇİNDEKİLER.....</b>	<b>iv</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>v</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>vi</b>
<b>SİMGELER VE KISALTMALAR DİZİNİ.....</b>	<b>vii</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. ZAMANLANDIRILMIŞ PETRİ AĞI (ZPA) MODELİ.....</b>	<b>7</b>
<b>3. ZPA'NIN ULAŞILABİLİRLİK KÜMESİ.....</b>	<b>15</b>
3.1. Ulaşılabilirlik Kümesinin Elde Edilmesi.....	15
<b>4. SİSTEM ÇIKMAZI.....</b>	<b>24</b>
<b>5. UYGULAMALAR.....</b>	<b>26</b>
<b>6. SONUÇLAR.....</b>	<b>31</b>
<b>KAYNAKLAR.....</b>	<b>33</b>
<b>EKLER.....</b>	<b>36</b>

**ŞEKİLLER DİZİNİ**

2.1.'de ele alınan ZPA.....	9
5.1.'de ele alınan ZPA.....	27
5.2.'de ele alınan ZPA.....	29

**ÇİZELGELER DİZİNİ**

2.1.'deki ZPA'nın [0,2] bz. aralığındaki işaretleme değişimi.....	13
---	----





- $\rho_0(\mathcal{M}, t)$  :  $t \in T$  geçişinin ateşlenmesini sonlandırıp,  $\mathcal{M}$  işaretleme vektöründe yeni belirtileri bu geçişin çıktı yerlerine ilave eden fonksiyon;  $\rho_0(\mathcal{M}, t) := \mathcal{M} + O_t$
- $RS$  :  $G$  ile gösterilen ZPA'nın  $[\tau_0, \tau_f]$  zaman aralığındaki ulaşılabilirlik kümesi
- $RSt$  :  $RS$  kümesindeki her bir işaretlemenin elde edilme anlarının tutulduğu küme
- $F_t$  :  $RS$  kümesindeki her bir işaretlemede, ateşlenebilir halde olan geçişlerin indislerinin tutulduğu küme
- $E(\mathcal{M})$  :  $\mathcal{M}$  işaretleme vektöründe ateşlenebilir olan geçişlerin kümesi
- $DM$  : Sistem çıkmazının meydana geldiği işaretleme vektörlerinin oluşturduğu küme
- $DMt$  :  $DM$  kümesindeki her bir işaretlemenin elde edilme anlarının tutulduğu küme

## 1. GİRİŞ

Günümüzde üretim, haberleşme, bilişim sistemleri ve daha birçok sistem teknolojideki gelişmelere paralel olarak hızla büyümekte, gelişim göstermekte ve bunlara bağlı olarak da gitgide daha karmaşık bir hal almaktadır. Üretim sistemleri, haberleşme ağları, vb. sistemler olay etkileşimli (event driven) sistemlerin kapsamına girmektedir. Bunun nedeni bu sistemlerde bir olayın gerçekleşmesinin başka bir olayın meydana gelmesine bağlı olmasıdır. Bu sistemlere kesikli olay sistemleri (discrete event systems) adı verilmektedir [Ho, 1992].

Birçok kesikli olay sisteminin verimliliğinin arttırılması, oluşabilecek hataların belirlenmesi ve bunların önlenmesi amacıyla çeşitli modelleme yöntemleri uzun yıllardır kullanılmaktadır. Bunlardan bazıları, Markov zincirleri, otomota modelleri, Petri ağları, sonlu durum makineleri, minimum - maksimum cebir modelleri ve genelleştirilmiş yarı Markov işlem modelleridir [Ho, 1992].

Petri ağı yapısı ilk kez Carl Adam Petri tarafından 1962'de ortaya konmuştur [Petri, 1962]. Petri ağı ele alınan sistemin grafiksel olarak modellenmesine olanak tanıyan ve oluşturulan grafiksel model ile ilgili matematiksel ifadelerin rahatça oluşturulabildiği etkili bir modelleme aracıdır. Bir Petri ağı temelde üç ana unsurdan meydana gelmektedir: yerler, geçişler ve bunlar arasındaki bağlantılar [Zhou ve DiCesare, 1993]. Petri ağı modellemesinin zamanlandırılmış (timed) Petri ağı (ZPA), renklendirilmiş (coloured) Petri ağı, zamanlandırılmış - renklendirilmiş (timed - coloured) Petri ağı, kontrol edilen (controlled) Petri ağı, hibrit (hybrid) Petri ağı, v.b. Petri ağı çeşitleri vardır [Desrochers ve Al-Jaar, 1995]. Bu çalışmada kesikli olay sistemlerinin modellenmesinde ve analizinde zamanlandırılmış Petri ağı modeli kullanılacaktır.

Bu çalışmada ele alınan zamanlandırılmış Petri ağlarının kapsamında ilk olarak farklı modeller incelenmiştir. Temelde iki farklı yaklaşım bulunmaktadır. Bunlardan ilki geçişlere zaman ifadelerinin atanmasıdır, diğeri de zaman ifadelerinin yerlere atanmasıdır [Bowden, 2000], [López-Mellado, 2002], [Wang, 1998]. Zaman ifadelerinin geçişlere atandığı Petri ağına geçişleri

zamanlandırılmış (timed transition, T-timed) Petri ağı adı verilmektedir. Zaman ifadelerinin yerlere atandığı Petri ağına ise yerleri zamanlandırılmış (timed place, P-timed) Petri ağı adı verilmektedir.

Geçişleri zamanlandırılmış Petri ağları kapsamında C. Ramchandani tarafından sunulan çalışmadaki, [Ramchandani, 1974], modelde zaman gecikmeleri her bir geçişe atanmış negatif olmayan gerçek sayılar şeklinde ifade edilmiştir. Her bir geçişe atanmış olan gerçek sayı o geçişin ateşlenme gecikmesi olarak adlandırılmıştır. Geçişlere atanmış olan bu ateşlenme gecikmelerine bağlı olarak bir geçişin ateşlenmesinin ZPA'nın işaretlemesinde iki basamaklı bir değişime neden olacağı ifade edilmiştir. İlk adımda ateşlenen geçişin girdi yerlerinden belirtilerin anlık olarak eksiltileceği, ikinci adım olarak ise bu geçişe ait ateşlenme gecikmesi kadar süre sonra bu geçişin çıktı yerlerinde yeni belirtilerin oluşturulacağı vurgulanmıştır. Bu modeli temel alan Zuberek tarafından sunulan çalışmalarda, ([Zuberek, 1991, 2001]), geçişlerin ateşlenmelerindeki beklemenin ne kadar olacağını ifade eden "ateşlenme zamanları"nın belirli (deterministic) veya rassal (stochastic) olabileceğinden bahsedilmiştir. Ayrıca ilgili çalışmalarda her bir geçişe belirli bir seçilme olasılığı atayan bir fonksiyonun bulunduğu belirtilmiştir. Bu modeli temel alan Chiola ve Ferscha tarafından sunulan çalışmada, [Chiola ve Ferscha, 1993], geçişlere gerçek sayılarla verilen zaman ifadelerinin yanında farklı öncelik (priority) değerlerinin atanmış olduğu ve ateşlenebilirlik değerlendirmesi yapılırken bu değerlerin göz önünde bulundurulduğu belirtilmiştir. Eğer bir geçişin girdi yerlerinde yeterli sayıda belirti bulunuyorsa (geçiş aktifleşmişse) ve bu anda bu geçişinkinden daha büyük öncelik değerine sahip başka bir geçiş aktifleşmiş olarak bulunmuyorsa, bu geçişin ateşlenmesine anlık olarak başlanacağı ifade edilmiştir. İlgili çalışmada ayrıca yerler ve geçişler arasındaki bağlantılara farklı ağırlık değerleri atanmıştır. ZPA'daki bağlantılara (oklara) farklı ağırlık değerlerinin atanmış olduğu diğer bir çalışma da Lefebvre ve El Moudni tarafından sunulan çalışmadır [Lefebvre ve El Moudni, 2001]. İlgili çalışmada, bir geçişe atanmış olan gerçek bir sayı ile verilen zaman gecikmesinin, aynı transfer elemanına ardısıra gelen iki olayı ayıran

gecikmeyi belirttiğinden bahsedilmiştir. Diğer çalışmalardan farklı olarak aktifleşen bir geçişin girdi yerlerinde belirtilerin, bu geçiş için ayrılmış olarak tutulacağı ifade edilmiştir. Bu ayrılma olayının geçiş aktifleşir aktifleşmez gerçekleşeceği ve bunun nedeninin Petri ağının azami hızda çalışması olduğu belirtilmiştir. Ateşlenmesine başlanan bir geçiş için ayrılmış halde bulunan belirtiler mevcut iken yeni bir geçişin aktifleşmesi için sadece hiçbir geçiş için ayrılmamış halde bulunan belirtilerin göz önüne alınacağı vurgulanmıştır. Ramchandani'nin yaklaşımını esas alan çalışmaların çoğunda, ([Buy ve Sloan, 1994], [Ling ve Schmidt, 1996], [Gaubert ve Mairesse, 1999]), ele alınan Petri ağının güvenli olduğu kabullenmesi yapılmıştır. Sınırlılık özelliğinin özel bir hali olan güvenlilik özelliğinde, Petri ağındaki yerlerde ulaşılan her bir işaretlemede en fazla bir tane belirti bulunabilmektedir. İlgili çalışmalarda bunun yanı sıra bağlantılara farklı ağırlık değerleri atanmamış ve tüm bağlantıların ağırlık değeri bir olarak alınmıştır. Bu kabullenmeler matematiksel ifadeleri basitleştirmek amacıyla yapılmıştır. Yapılan güvenlilik kabullenmesi sonucunda zaman, geçişlerin ateşlenebilirliğini fazla etkilememektedir. Bu durumda, herhangi bir geçişin ateşlenmesi devam ederken başka bir geçiş ateşlenmemektedir. Ayrıca, bir geçişin belirli bir anda birden fazla ateşlenebilir olması durumu bu kabullenmeyle önlenmektedir.

Geçişleri zamanlandırılmış Petri ağları kapsamında Merlin tarafından sunulan çalışmadaki, [Merlin, 1974], modelde ise her bir geçişe negatif olmayan gerçek sayılarla sınırlandırılmış bir zaman aralığı atanmıştır ve bu aralık statik ateşlenme aralığı olarak adlandırılmıştır. Her bir geçişe belirli bir gerçek sayı değil de bir gecikme aralığı atanmış olduğundan bu modelin gerçek zamanlı uygulamaların modellenmesi konusunda daha avantajlı olduğundan bahsedilmiştir [Berthomieu ve Menasche, 1983]. Bu modeli temel alan, Janneck tarafından sunulan çalışmada, [Janneck, 1998], belirtilerin zaman bilgisini içerdiği ve girdi belirtilerine ait değerlerden çıktı belirtilerine ait değerlerin hesaplandığı fonksiyonların geçişlerle ilişkilendirilmiş olduğu kabul edilmiştir. ZPA'da her bir geçişe belirli bir zaman aralığı atanmış olduğu ifade edilmiştir. İlgili çalışmadaki ifadelerde, bağlantılara atanan

farklı ağırlık değerlerinin bulunabileceğinden bahsedilmiştir. Ayrıca, ilgili çalışmada bir geçişin girdi yerlerindeki belirtilerle birden fazla eş zamanlı olarak aktif hale getirilebileceği ve buna bağlı olarak bu geçişin birden fazla birbirinden bağımsız ateşlenmesinin gerçekleşebileceği belirtilmiştir. Bu birbirinden bağımsız ateşlenmeler arasında çok küçük bir zaman bulunacağı ve hatta bazen bu ateşlenmelerin aynı anda gerçekleşeceği ifade edilmiştir. Lilius tarafından sunulan Merlin'in modelini, [Merlin, 1974], temel alan çalışmada, [Lilius, 1998], ZPA'larda sistemin durumunun aynı zamanda zamanlama bilgisini de içermesi gerektiğinden bahsedilmiştir. Bu bilginin kullanılabilir haldeki bir geçişin kullanılabilir hale geçtiği andan içinde bulunan zamana kadar geçen süreyi ifade eden bir saat fonksiyonu ile verildiği belirtilmiştir. İlgili çalışmada Merlin'in modelinin olayların kesin gerçekleşme sürelerinin bilinmediği durumlarda eksik (imperfect) zamanlama bilgisini modellemeyi mümkün kıldığı vurgulanmıştır. Koutsoukos ve ark. tarafından sunulan çalışmada, [Koutsoukos ve ark., 1998], bir geçişin ateşlenmesinin belirli bir zaman aralığı boyunca gerçekleştiğinden ve bu aralığın o geçişin tutulma zamanı (holding time) olarak adlandırıldığından bahsedilmiştir. Bu yönde ortaya konmuş olan ilgili çalışmada, bu zaman aralıklarının sistemin tasarımcısı tarafından programlanan belirli koşullar sağlandığında Petri ağındaki geçişleri ateşlenmeye zorlayan lokal zamanlayıcılar ile kontrol edilebildiği programlanabilir zamanlandırılmış Petri ağları ele alınmıştır.

Freedman tarafından sunulan çalışmada, [Freedman, 1991], zaman ifadelerinin geçişlere atanmış olduğu bu iki farklı yaklaşıma da yer vermiştir. İlgili çalışmada ele alınan Petri ağlarında yerler ve geçişler arasındaki bağlantıların ağırlık değerleri bir olarak alınmıştır. Ramchandani'nin yaklaşımı, [Ramchandani, 1974], ele alındığında geçişlere belirli bir bekleme süresi atandığı ifade edilmiştir. Merlin'in yaklaşımı, [Merlin, 1974], ele alındığında ise her bir geçişe bir bekleme aralığının atanmış olduğu vurgulanmıştır. Ramchandani'nin modelinde ifade edilen bir olaya ait belirli bekleme süresinin Merlin'in modelinde ilgili geçişe ait bekleme aralığının sınırları birbirine eşit alınarak kolaylıkla ifade edilebileceği, bu nedenle de Merlin'in modelinin daha

genel bir zaman modeli olduğu belirtilmiştir. İlgili çalışmada, bunların yanında Merlin'in yaklaşımına yakın olan Berthomieu-Menasche'nin yaklaşımına da değinilmiştir.

Merlin ve Ramchandani'den farklı olarak Sifakis tarafından sunulan çalışmadaki, [Sifakis, 1977], modelde zaman parametrelerinin yerlere atanmış olduğu görülmüştür. Bu modelde yerlere negatif olmayan gerçek sayılar atayan bir fonksiyon bulunduğu bahsedilmiştir. Her bir yere atanmış olan gerçek sayı o yerin bekleme zamanı olarak adlandırılmıştır. Zaman parametrelerinin geçişlere atanmış olduğu model ile zaman parametrelerinin yerlere atanmış olduğu modelin rahatlıkla birbirine dönüştürülebileceğinin bizzat Sifakis'in kendisi tarafından ifade edildiğinden bahsedilmiştir [Barad, 1998], [Bowden, 2000].

Bunlardan farklı olarak Tsai ve ark. tarafından sunulan çalışmada, [Tsai ve ark., 1995], zaman sınırlamalı Petri ağlarından (Timing Constraint Petri Nets, TCPN) bahsedilmektedir. Zaman sınırlamalı Petri ağlarının Ramchandani'nin modeli ile Merlin'in modelinin bir sentezi olduğu ifade edilmiştir. Yerlerle ve geçişlerle ilişkilendirilen zaman sınırlamalarının bulunduğu ifade edilmiştir. İlgili çalışmada, Coolahan ve Roussopoulos tarafından sunulan çalışmada, [Coolahan ve Roussopoulos, 1983], yaklaşım olarak yerlere bekleme süreleri atandığına değinilmiş, ancak bu çalışmada geçişlere ateşlenme gecikmelerinin atandığı yaklaşımın esas alındığı belirtilmiştir. Huifang ve Yushun tarafından sunulan çalışmada da, [Huifang ve Yushun, 2002], benzer şekilde zamanlandırılmamış Petri ağına, yerlere ve geçişlere ait olabilen, biri minimum diğeri ise maksimum sınırlamayı belirten sıralı ikililerin oluşturduğu küme ile geçişlere atanmış olan ateşlenme sürelerinin kümesinin eklenmesiyle zaman sınırlamalı Petri ağları oluşturulmuş ve incelenmiştir.

Zamanlandırılmış - renklendirilmiş Petri ağları için, Kobayashi ve ark. tarafından bir yaklaşım geliştirilmiştir [Kobayashi ve ark., 2001]. İlgili çalışmada, notasyonu basitleştirmek amacıyla yapılan bazı kabullenmeler olduğu ifade edilmiştir. Bu kabullenmelerden ilki, Petri ağının güvenli olduğu kabullenmesidir. Diğer kabullenme ise yerler ve geçişler arasındaki bağlantıların

ağırlık değerlerinin bir olarak alınmasıdır.

Bu tez çalışmasında zaman ifadelerinin geçişlere atanmış olduğu yaklaşım benimsenecektir. Bunun nedeni, geçişlerin Petri ağlarında bir durumdan diğer bir duruma ulaşmakta kullanılan olayları ifade etmesi ve gerçek zamanlı uygulamalarda her bir olayın gerçekleşmesinin belirli bir zaman almasıdır. Geçişleri zamanlandırılmış Petri ağı modellerinden Ramchandani'nin modeline uygun olarak ateşlenme gecikmelerinin her bir geçiş için sabit bir değer olduğu kabul edilecektir. Bu çalışmada, diğer çalışmalardan farklı olarak, Petri ağının sınırlı olduğu kabullenmesi yapılacaktır. Buna göre Petri ağındaki yerlerdeki belirti sayıları ulaşılan her bir işaretleme için bir doğal sayı ile sınırlı olmak zorundadır. Bunun sonucunda, herhangi bir geçiş birden fazla peş peşe ateşlenebilmekte ve devam eden ateşlenmeler mevcut iken yeni geçişlerin ateşlenmesine olanak verilmektedir.

Bölüm 2'de bu tez çalışmasında ele alınan zamanlandırılmış Petri ağı modeli ve bununla ilgili matematiksel ifadeler verilecektir. Bölüm 3'te ZPA'nın ulaşılabilirlik kümesinin elde edilmesi detaylı olarak anlatılacaktır. Bölüm 4'te ZPA'da sistem çıkmazının nasıl oluştuğu ve sistem çıkmazının oluştuğu işaretlemelerin nasıl tespit edildiği anlatılacaktır. Oluşturulan MATLAB programı kullanılarak farklı ZPA'lar üzerinde gerçekleştirilen ulaşılabilirlik kümesinin tespiti ve sistem çıkmazı analizlerine dair uygulamalar Bölüm 5'te verilecektir. Uygulamalarda elde edilen verilerin değerlendirilmesi neticesinde ulaşılan sonuçlar Bölüm 6'da verilecektir.



## 2. ZAMANLANDIRILMIŞ PETRİ AĞI (ZPA) MODELİ

Zamanlandırılmış Petri Ağı  $G(P, T, N, O, m_0, \tau_0, D, \Lambda)$  gösterimi ile ifade edilmektedir.  $P$ , yerlerin kümesini,  $T$ , geçişlerin kümesini ( $P \cap T = \emptyset$  ve  $P \cup T \neq \emptyset$ ),  $N : P \times T \rightarrow \mu =: \{0, 1\}$  geçişlerden yerlere doğru olan bağlantıları içeren girdi matrisini ve  $O : P \times T \rightarrow \mu$  yerlerden geçişlere doğru olan bağlantıları içeren çıktı matrisini göstermektedir. ZPA'daki her bir yerde başlangıç anında,  $\tau_0$ , bulunan belirti sayısı,  $m_0 : P \rightarrow \mathcal{N}$  başlangıç işaretlemesi ile ifade edilmektedir.  $D : T \rightarrow \mathcal{R}^+$  geçişlerin ateşlenme gecikmelerinin kümesidir ve  $D := \{d_{t_i} | i \in \{1, 2, \dots, |T|\}\}$  şeklinde tanımlanmaktadır.  $|\cdot|$  ile  $(\cdot)$  kümesinin eleman sayısı ifade edilmektedir.  $\Lambda : T \rightarrow \mathcal{R}^+$  geçişlerin bekleme sürelerinin kümesidir ve  $\Lambda := \{\lambda_{t_i} | i \in \{1, 2, \dots, |T|\}\}$  şeklinde tanımlanmaktadır. Eğer  $t \in T$  geçişi  $\tau$  anında birden fazla peş peşe ateşlenebilmekte ise bu geçişin ardışıl ateşlenmelerini birbirinden ayırmak için  $\lambda_t \in \Lambda$  ile gösterilen bekleme süresi kullanılır.

ZPA'nın herhangi bir  $\tau$  anındaki durumu,  $M(\tau) : P \rightarrow \mathcal{N}$  işaretleme vektörü ile ifade edilmektedir. Burada  $\tau$  genel zamanı göstermektedir. Petri ağındaki bir  $p \in P$  yerinde  $\tau$  anında bulunan belirti sayısı  $M(p, \tau)$  ifadesi ile verilmektedir.

Petri ağındaki herhangi bir  $t \in T$  geçişi için ateşlenebilirlik koşulu şu şekilde ifade edilmektedir:

$$M(p, \tau) \geq 1, \forall p \in \bullet t := \{p \in P \mid N(p, t) = 1\} \quad (2.1)$$

$\bullet t$ ,  $t$  geçişine girdi olarak bağlı olan yerlerin kümesini göstermektedir. Herhangi bir  $t \in T$  geçişi,  $\bullet t$  kümesinin elemanlarındaki belirti sayısına bağlı olarak birden fazla peş peşe ateşlenebilmektedir. Bir  $t \in T$  geçişinin  $\tau$  anında kaç kez ateşleneceği,  $m_t(\tau) : T \rightarrow \mathcal{N}$  ile gösterilen, ateşlenebilirlik derecesi ile ifade edilmektedir.  $t$  geçişinin  $\tau$  anındaki ateşlenebilirlik derecesi şu şekilde belirlenmektedir:

$$m_t(\tau) = \min_{p \in \bullet t} \left( \text{ceil} \left( \frac{M(p, \tau)}{\sum_{\hat{t} \in T} N(p, \hat{t})} \right) \right), \forall t \in T, \forall p \in \bullet t \quad (2.2)$$

Burada,  $ceil(\cdot)$ , içindeki gerçek sayıyı en yakın bir üst tamsayıya yuvarlayan fonksiyonu göstermektedir (örneğin,  $ceil(1,2) = 2$ ,  $ceil(1,0) = 1$ ).

Buradaki ifadeye göre bir  $t \in T$  geçişinin ateşlenebilirlik derecesini bulmak için ilk olarak bu geçişe girdi olarak bağlı olan yerler tespit edilir. Ardından  $t \in T$  geçişinin her bir girdi yerine çıktı olarak bağlı olan bütün geçişlerin sayısı tespit edilir. Daha sonra her bir yerdeki belirti sayısı yine her bir yer için elde edilen bu sayıya bölünür. Bu bölme işlemleri sonucunda elde edilen değerlere tek tek  $ceil$  fonksiyonu uygulanır. Sonuçta elde edilen tüm değerlerin minimumu belirlenir. Bu değer bize  $t \in T$  geçişinin ateşlenebilirlik derecesini vermektedir.

Bir  $t \in T$  geçişinin ateşlenebilirlik koşulunu sağladığı an  $e_t$  ile, ateşlenme anı ise  $f_t$  ile gösterilmektedir. Bu iki zaman arasında  $f_t \geq e_t$  şeklinde bir ilişki bulunmaktadır. Bu çalışmada,  $f_t = e_t$  olarak alınmaktadır.

Eğer herhangi bir  $t \in T$  geçişi için  $m_t(\tau) > 1$  ise,  $t$  geçişi  $\tau$  anında  $m_t(\tau)$  kere peş peşe ateşlenebilmektedir.  $f_t^0 = e_t$  olmak üzere, birden fazla ateşlenebilir halde bulunan  $t \in T$  geçişinin ateşlenme anları aşağıdaki ifade ile verilmektedir:

$$f_t^j = f_t^{j-1} + \lambda_t, j \in \{1, 2, \dots, m_t(\tau)\} \quad (2.3)$$

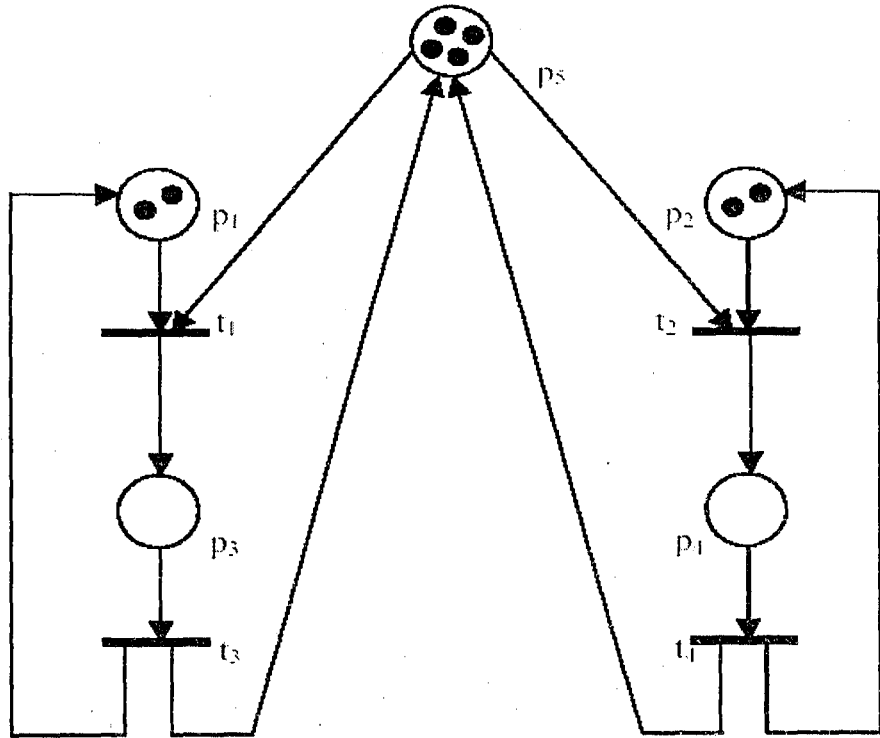
Eğer ateşlenebilirlik koşulunu sağlayan bir  $t \in T$  geçişi  $f_t$  ateşlenme anında ateşlenirse,  $[f_t, \tau]$  zaman aralığında işaretlemede şu şekilde bir değişim meydana gelmektedir:

$$M(p, \tau) = M(p, f_t) + \sum_{\hat{t} \in \theta_t(\tau)} \left[ \sum_{k=0}^{m_{\hat{t}}(\tau)-1} \left\{ \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}} + d_{\hat{t}}))O(p, \hat{t}) - \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}}))N(p, \hat{t}) \right\} \right], t \in T, \forall p \in P, \tau \geq f_t \quad (2.4)$$

Eğer  $y \geq 0$  ise  $\Phi(y) = 1$  (aksi durumda  $\Phi(y) = 0$ ) olmaktadır ( $y \in \mathcal{R}$ ). Burada,  $\theta_t(\tau) := \{t\} \cup \alpha_t(\tau)$ ,  $\alpha_t(\tau) := \{\hat{t} \in T | \hat{t}, [f_{\hat{t}}, \tau]$  aralığında ateşlenebilirlik koşulunu sağlamaktadır} şeklinde tanımlanmaktadır.

### Örnek 2.1

Şekil 2.1.'deki ZPA'da  $D = [3 \ 2 \ 1 \ 2]$  birim zaman (bz.) olarak ve  $\Lambda = [0.2 \ 0.1 \ 0.3 \ 0.4]$  bz. olarak verilmektedir. Bu örnek ZPA'da  $[0, 2]$  bz. aralığında, işaretleme vektörlerinin elde edilmesi incelenecektir. Başlangıç işaretlemesine,  $m_0 = [2 \ 2 \ 0 \ 0 \ 4]^T$ , göre  $\tau_0 = 0$  bz. anında ateşlenebilir olan geçişler  $t_1$  ve  $t_2$ 'dir.  $t_1$  ve  $t_2$  geçişleri için girdi ve çıktı yerlerinin kümeleri,  $\bullet t_1 = \{p_1, p_5\}$ ,  $\bullet t_2 = \{p_2, p_5\}$ ,  $t_1 \bullet = \{p_3\}$  ve  $t_2 \bullet = \{p_4\}$  şeklinde elde edilmektedir.



Şekil 2.1. Örnek 2.1'de ele alınan ZPA

$\tau_0 = 0$  anında  $t_1$  ve  $t_2$  geçişlerinin ateşlenebilirlik dereceleri,

$$m_{t_1}(\tau_0) = \min(\text{ceil}(M(p_1, \tau_0) / 1), \text{ceil}(M(p_5, \tau_0) / 2)) = \min(2, 2) = 2$$

$$m_{t_2}(\tau_0) = \min(\text{ceil}(M(p_2, \tau_0) / 1), \text{ceil}(M(p_5, \tau_0) / 2)) = \min(2, 2) = 2$$

olarak elde edilmektedir. Yani  $\tau_0$  anında,  $t_1$  ve  $t_2$  geçişleri üst üste iki kez ateşlenebilir. Bunların yanında

$$\theta_{t_1}(\tau_0) = \theta_{t_2}(\tau_0) = \{t_1\} \cup \{t_2\} = \{t_1, t_2\}$$

olmaktadır. Dolayısıyla  $\tau_0$  anında  $t_1$  ve  $t_2$  geçişleri ateşlenebilir.

Ateşlenebilirlik koşulunu sağlayan geçişlerin  $\tau_0$  anında ateşlenmesiyle birlikte ZPA'daki yerlere ait işaretlemelerde meydana gelen değişimi (2.4)'teki ifadeye göre inceleyelim:

$p_1$  yeri için  $M(p_1, \tau)$ , (2.4)'teki ifadeye göre;

$$M(p_1, \tau) = M(p_1, \tau_0) + \sum_{\hat{t} \in \{t_1, t_2\}} \left[ \sum_{k=0}^{2-1} \left\{ \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}} + d_{\hat{t}}))O(p_1, \hat{t}) - \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}}))N(p_1, \hat{t}) \right\} \right]$$

$$M(p_1, \tau) = M(p_1, \tau_0) + \left[ [\Phi(\tau - (\tau_0 + 3))O(p_1, t_1) - \Phi(\tau - \tau_0)N(p_1, t_1)] \right. \\ + [\Phi(\tau - (\tau_0 + 0.2 + 3))O(p_1, t_1) - \Phi(\tau - (\tau_0 + 0.2))N(p_1, t_1)] \\ + [\Phi(\tau - (\tau_0 + 2))O(p_1, t_2) - \Phi(\tau - \tau_0)N(p_1, t_2)] + \\ \left. [\Phi(\tau - (\tau_0 + 0.1 + 2))O(p_1, t_2) - \Phi(\tau - (\tau_0 + 0.1))N(p_1, t_2)] \right]$$

şeklinde elde edilmektedir.

Eğer  $\tau \in [0, 0.2)$  bz. ise,

$$M(p_1, \tau) = 2 + \left[ [0 - 1] + [0 - 0] + [0 - 0] + [0 - 0] \right] = 2 - 1 = 1$$

olarak elde edilir. Böylece  $t_1$  geçişi bir kez ateşlenmiş olmaktadır.

Eğer  $\tau \in [0.2, 2]$  bz. ise,

$$M(p_1, \tau) = 2 + \left[ [0 - 1] + [0 - 1] + [0 - 0] + [0 - 0] \right] = 2 - 2 = 0$$

olarak elde edilir. Bu durumda ise  $t_1$  geçişi iki kez ateşlenmiş olmaktadır.

$p_2$  yeri için  $M(p_2, \tau)$ , (2.4)'teki ifadeye göre;

$$M(p_2, \tau) = M(p_2, \tau_0) + \sum_{\hat{t} \in \{t_1, t_2\}} \left[ \sum_{k=0}^{2-1} \left\{ \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}} + d_{\hat{t}}))O(p_2, \hat{t}) - \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}}))N(p_2, \hat{t}) \right\} \right]$$

$$\begin{aligned}
M(p_2, \tau) = & M(p_2, \tau_0) + \left[ [\Phi(\tau - (\tau_0 + 3))O(p_2, t_1) - \Phi(\tau - \tau_0)N(p_2, t_1)] \right. \\
& + [\Phi(\tau - (\tau_0 + 0.2 + 3))O(p_2, t_1) - \Phi(\tau - (\tau_0 + 0.2))N(p_2, t_1)] \\
& + [\Phi(\tau - (\tau_0 + 2))O(p_2, t_2) - \Phi(\tau - \tau_0)N(p_2, t_2)] + \\
& \left. [\Phi(\tau - (\tau_0 + 0.1 + 2))O(p_2, t_2) - \Phi(\tau - (\tau_0 + 0.1))N(p_2, t_2)] \right]
\end{aligned}$$

şeklinde elde edilmektedir.

Eğer  $\tau \in [0, 0.1)$  bz. ise,

$$M(p_2, \tau) = 2 + \left[ [0 - 0] + [0 - 0] + [0 - 1] + [0 - 0] \right] = 2 - 1 = 1$$

olarak elde edilir. Böylece  $t_2$  geçişi bir kez ateşlenmiş olmaktadır.

Eğer  $\tau \in [0.1, 2]$  bz. ise

$$M(p_2, \tau) = 2 + \left[ [0 - 0] + [0 - 0] + [0 - 1] + [0 - 1] \right] = 2 - 2 = 0$$

olarak elde edilir. Bu durumda  $t_2$  geçişi iki kez ateşlenmiş olmaktadır.

$p_5$  yeri için  $M(p_5, \tau)$ , (2.4)'teki ifadeye göre;

$$\begin{aligned}
M(p_5, \tau) = & M(p_5, \tau_0) + \sum_{i \in \{t_1, t_2\}} \left[ \sum_{k=0}^{2-1} \left\{ \Phi(\tau - (f_i + k\lambda_i + d_i))O(p_5, \hat{t}) - \right. \right. \\
& \left. \left. \Phi(\tau - (f_i + k\lambda_i))N(p_5, \hat{t}) \right\} \right]
\end{aligned}$$

$$\begin{aligned}
M(p_5, \tau) = & M(p_5, \tau_0) + \left[ [\Phi(\tau - (\tau_0 + 3))O(p_5, t_1) - \Phi(\tau - \tau_0)N(p_5, t_1)] \right. \\
& + [\Phi(\tau - (\tau_0 + 0.2 + 3))O(p_5, t_1) - \Phi(\tau - (\tau_0 + 0.2))N(p_5, t_1)] \\
& + [\Phi(\tau - (\tau_0 + 2))O(p_5, t_2) - \Phi(\tau - \tau_0)N(p_5, t_2)] + \\
& \left. [\Phi(\tau - (\tau_0 + 0.1 + 2))O(p_5, t_2) - \Phi(\tau - (\tau_0 + 0.1))N(p_5, t_2)] \right]
\end{aligned}$$

şeklinde elde edilmektedir.

Eğer  $\tau \in [0, 0.1)$  bz. ise,

$$M(p_5, \tau) = 4 + \left[ [0 - 1] + [0 - 0] + [0 - 1] + [0 - 0] \right] = 4 - 2 = 2$$

olarak elde edilir. Böylece  $t_1$  ve  $t_2$  geçişleri birer kez ateşlenmiş olmaktadır.

Eğer  $\tau \in [0.1, 0.2)$  bz. ise,

$$M(p_5, \tau) = 4 + \left[ [0 - 1] + [0 - 0] + [0 - 1] + [0 - 1] \right] = 4 - 3 = 1$$

olarak elde edilir. Bu durumda  $t_1$  geçişi bir kez ve  $t_2$  geçişi de iki kez ateşlenmiş olmaktadır.

Eğer  $\tau \in [0.2, 2]$  bz. ise,

$$M(p_5, \tau) = 4 + \left[ [0 - 1] + [0 - 1] + [0 - 1] + [0 - 1] \right] = 4 - 4 = 0$$

olarak elde edilir. Bu durumda ise  $t_1$  ve  $t_2$  geçişleri ikişer kez ateşlenmiş olmaktadır.

$p_3$  yeri için  $M(p_3, \tau)$ , (2.4)'teki ifadeye göre;

$$M(p_3, \tau) = M(p_3, \tau_0) + \sum_{i \in \{t_1, t_2\}} \left[ \sum_{k=0}^{2-1} \left\{ \Phi(\tau - (f_i + k\lambda_i + d_i))O(p_3, \hat{t}) - \Phi(\tau - (f_i + k\lambda_i))N(p_3, \hat{t}) \right\} \right]$$

$$\begin{aligned} M(p_3, \tau) = & M(p_3, \tau_0) + \left[ [\Phi(\tau - (\tau_0 + 3))O(p_3, t_1) - \Phi(\tau - \tau_0)N(p_3, t_1)] \right. \\ & + [\Phi(\tau - (\tau_0 + 0.2 + 3))O(p_3, t_1) - \Phi(\tau - (\tau_0 + 0.2))N(p_3, t_1)] \\ & + [\Phi(\tau - (\tau_0 + 2))O(p_3, t_2) - \Phi(\tau - \tau_0)N(p_3, t_2)] + \\ & \left. [\Phi(\tau - (\tau_0 + 0.1 + 2))O(p_3, t_2) - \Phi(\tau - (\tau_0 + 0.1))N(p_3, t_2)] \right] \end{aligned}$$

şeklinde elde edilmektedir.

Eğer  $\tau \in [0, 3)$  bz. ise,

$$M(p_3, \tau) = 0 + \left[ [0 - 0] + [0 - 0] + [0 - 0] + [0 - 0] \right] = 0 + 0 = 0$$

olarak elde edilir. Böylece  $[0, 3)$  aralığında ateşlenmesi tamamlanan geçiş bulunmamaktadır.

Eğer  $\tau = 3$  bz. ise,

$$M(p_3, \tau) = 0 + \left[ [1 - 0] + [0 - 0] + [0 - 0] + [0 - 0] \right] = 0 + 1 = 1$$

olarak elde edilir. Bu durumda  $t_1$  geçişi ateşlenmesini tamamlamaktadır ve böylece  $t_3$  geçişi ateşlenebilirlik koşulunu sağlamaktadır.

$p_4$  yeri için  $M(p_4, \tau)$ , (2.4)'teki ifadeye göre;

$$M(p_4, \tau) = M(p_4, \tau_0) + \sum_{\hat{t} \in \{t_1, t_2\}} \left[ \sum_{k=0}^{2-1} \left\{ \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}} + d_{\hat{t}}))O(p_4, \hat{t}) - \Phi(\tau - (f_{\hat{t}} + k\lambda_{\hat{t}}))N(p_4, \hat{t}) \right\} \right]$$

$$\begin{aligned} M(p_4, \tau) = & M(p_4, \tau_0) + \left[ [\Phi(\tau - (\tau_0 + 3))O(p_4, t_1) - \Phi(\tau - \tau_0)N(p_4, t_1)] \right. \\ & + [\Phi(\tau - (\tau_0 + 0.2 + 3))O(p_4, t_1) - \Phi(\tau - (\tau_0 + 0.2))N(p_4, t_1)] \\ & + [\Phi(\tau - (\tau_0 + 2))O(p_4, t_2) - \Phi(\tau - \tau_0)N(p_4, t_2)] + \\ & \left. [\Phi(\tau - (\tau_0 + 0.1 + 2))O(p_4, t_2) - \Phi(\tau - (\tau_0 + 0.1))N(p_4, t_2)] \right] \end{aligned}$$

şeklinde elde edilmektedir.

Eğer  $\tau \in [0, 2)$  bz. ise,

$$M(p_4, \tau) = 0 + \left[ [0 - 0] + [0 - 0] + [0 - 0] + [0 - 0] \right] = 0 + 0 = 0$$

olarak elde edilir. Böylece  $[0, 2)$  aralığında ateşlenmesi tamamlanan geçiş bulunmamaktadır.

Eğer  $\tau = 2$  bz. ise

$$M(p_4, \tau) = 0 + \left[ [0 - 0] + [0 - 0] + [1 - 0] + [0 - 0] \right] = 0 + 1 = 1$$

olarak elde edilir. Bu durumda  $t_2$  geçişi ateşlenmesini tamamlamaktadır ve böylece  $t_4$  geçişi ateşlenebilirlik koşulunu sağlamaktadır. Şekil 2.1.'deki ZPA için  $[0, 2]$  bz. aralığında elde edilen değerler Çizelge 2.1.'de verilmektedir.

Çizelge 2.1. Şekil 2.1.'deki ZPA'nın  $[0, 2]$  bz. aralığındaki işaretleme değişimi

İşaretleme vektörü	$\tau$
$[2 \ 2 \ 0 \ 0 \ 4]^T$	0 bz.*
$[1 \ 1 \ 0 \ 0 \ 2]^T$	$[0, 0.1)$ bz.*
$[1 \ 0 \ 0 \ 0 \ 1]^T$	$[0.1, 0.2)$ bz.
$[0 \ 0 \ 0 \ 0 \ 0]^T$	$[0.2, 2)$ bz.
$[0 \ 0 \ 0 \ 1 \ 0]^T$	2 bz.

\* :  $\tau_0 = 0$  bz. başlangıç anında mevcut olan  $m_0 = [2 \ 2 \ 0 \ 0 \ 4]^T$  başlangıç işaretlemesinde ateşlenebilir olan  $t_1$  ve  $t_2$  geçişleri ateşlendiğinde,  $[1 \ 1 \ 0 \ 0 \ 2]^T$  işaretleme vektörü de 0 bz. anında elde edilmektedir. Bu nedenle tabloda ikinci işaretleme vektörünün mevcut olduğu aralığın alt sınırı olan 0 bz. anı bu aralığa dahil edilmektedir.



### 3. ZPA'NIN ULAŞILABİLİRLİK KÜMESİ

Bu bölümde, ZPA için ulaşılabilirlik kümesinin nasıl elde edileceği anlatılacaktır. Ulaşılabilirlik kümesi  $m_0$  başlangıç işaretlemesinden başlayarak elde edilebilir bütün işaretlemeleri içeren kümedir.  $G$  ile gösterilen ZPA'nın ulaşılabilirlik kümesi  $R(G, \tau_f)$  ile gösterilmektedir. Buna göre ulaşılabilirlik kümesi için analizin başlangıç ve sona erme zamanları belirleyici olmaktadır. Ulaşılabilirlik kümesi Petri ağı ile ilgili özelliklerin analizlerinde kullanılmaktadır [Zhou ve DiCesare, 1993].

#### 3.1. Ulaşılabilirlik Kümesinin Elde Edilmesi

Ulaşılabilirlik kümesi meydana getirilirken  $\tau_0$  anındaki başlangıç işaretlemesinden başlayarak  $\tau_f$  anına kadar, ulaşılan her bir yeni işaretleme için sırasıyla şu işlemler gerçekleştirilir:

- 1) Hangi geçişlerin kaç kez ateşlenebilir olduğu tespit edilir.
- 2) Ateşlenebilir halde bulunan geçişler ateşlenir.
- 3) Başlatılan ateşlenmeler başlama sırasına göre tamamlanır.

Eğer ZPA'da ulaşılan bir işaretlemede birden fazla geçişin ortak bir girdi yerinde ilgili tüm geçişlerin ateşlenmesine yetecek sayıda belirti bulunmuyor ise çelişki oluşmaktadır. Bu durumda, geçiş veya geçişlerin bazılarının ateşlenmek için seçilmesi gerekmektedir. Bu çalışmada, her bir geçişin ayrı ayrı seçildiği bütün olası durumlar ele alınmaktadır. Bu sayede ulaşılabilirlik kümesinin bütün olası işaretlemeleri içermesi sağlanmaktadır.

Ulaşılabilirlik kümesi oluşturulurken Ek-B ve Ek-C'de verilen algoritmalar kullanılmaktadır. İşlem kolaylığının sağlanması ve ulaşılabilirlik kümesinin elde edilmesi ile ilgili algoritmanın çok uzun ifadelerle verilmesinin önlenmesi amacıyla ilgili algoritma UK-1 ve UK-2 gibi iki farklı algoritma kullanılarak oluşturulmuştur.

UK-1 algoritması için  $G$  ile gösterilen ZPA'yı tanımlayan bilgilerin (yerlerin kümesi, geçişlerin kümesi, girdi matrisi, çıktı matrisi, başlangıç işaretleme, başlangıç zamanı, ateşlenme gecikmelerinin kümesi ve geçişlerin

bekleme sürelerinin kümesi) girdi olarak girilmesi gerekmektedir. UK-1 algoritmasında başlangıç işaretlemesinde hangi geçişlerin kaç kez ateşlenebilir oldukları tespit edilmektedir. Ardından  $\Lambda$  kümesinin elemanları kullanılarak başlangıç işaretlemesinde birden fazla ateşlenebilir olan geçişlerin ilk ateşlenmelerini takip eden ateşlenmelerinin hangi zamanlarda başlayacağı bulunur. Ateşlenebilir halde olan tüm geçişlerin ilk ateşlenmeleri  $\tau_0$  başlangıç anında anlık olarak başlatılır ve buna bağlı olarak elde edilen işaretleme vektörü  $m_0$  işaretleme vektörünün yanına ulaşılabilirlik kümesinin ikinci elemanı olarak ilave edilir. Ardından “*adım*” (ilerleyen sayfalarda detaylı olarak verilecektir) kadarlık zaman artırımı yapılır ve birden fazla ateşlenebilir olan geçişlerin ikinci ve daha sonraki ateşlenme anlarına ulaşıldığında ilgili geçişlerin ateşlenmeleri başlatılır. Böylece bu geçişlerin girdi yerlerinden tekrar belirtiler eksiltir ve elde edilen her bir yeni işaretleme vektörü ulaşılabilirlik kümesine ilave edilir. Bu işlem başlangıç işaretlemesinde ateşlenebilirlik derecesi en büyük olan, yani en fazla sayıda ateşlenebilen geçişin son ateşlenmesi tamamlanmaya kadar devam eder.

UK-2 algoritması için  $G$  ile gösterilen ZPA’yı tanımlayan bilgilerin, analizin sona erme zamanının,  $BT$ ,  $ET$ ,  $RS$ ,  $RSt$  ve  $F_t$  kümelerinin,  $BT$  kümesinin içindeki maksimum zaman değerinin ve bu andaki işaretleme vektörünün,  $ET$  kümesinin içindeki minimum zaman değerinin ve zaman artırımını gösteren *adım* değerinin girdi olarak girilmesi gerekmektedir. UK-2 algoritmasında ise geçişlerin ateşlenmelerinin başlama anlarına her bir geçişe ait ateşlenme gecikmesi süresinin ilave edilmesiyle elde edilen ateşlenme tamamlanma anlarından minimumuna ulaşıldığında ilgili geçişin ateşlenmesi tamamlanır ve belirtiler ilgili geçişin çıktı yerlerinde oluşturulur. Daha sonra tamamlanan ateşlenme sonucunda ateşlenebilir hale geçen geçiş veya geçişlerin bulunup bulunmadığı kontrol edilir. Ateşlenebilir hale geçen geçişler mevcut ise bunların ateşlenmeleri başlatılır ve başlama anına ilgili geçişin ateşlenme gecikmesi ilave edilerek ateşlenmenin sona erme zamanı elde edilir. Bundan sonra yine en erken tamamlanması gereken ateşlenme tamamlanır. Buraya kadar belirtilen işlemler bir döngü halinde analizin sona erme zamanına,  $\tau_f$ , kadar gerçekleştirilir. Bu işlemler esnasında elde edilen tüm farklı işaretleme vektörleri ulaşılabilirlik kümesine ilave edilmektedir.

Ulaşılabilirlik kümesinin oluşturulması sırasında kullanılan algoritmalarda  $BT$ ,  $ET$  ve  $HT$  ile gösterilen kümeler önemli bir rol oynamaktadır.  $BT$ ,  $ET$ ,  $HT$  kümeleri her bir geçişe karşılık gelen vektörlerden oluşmuş kümelerdir. Başlangıçta bu kümeler  $BT = \phi := \{ [ [ ] \dots [ ] ] \}$ ,  $ET = \phi$  ve  $HT = \phi$  olmaktadır. Burada,  $\phi$ , içinde bulunan tüm vektörlerin boş olduğu küneyi göstermektedir.  $BT$  kümesinde içinde bulunulan anda ateşlenebilir olan geçişler için ateşlenme anları tutulur. Eğer bazı geçişler birden fazla ateşlenebilmekte ise, bu geçişlerin ilk ateşlenmeden sonraki ateşlenme anları  $\Lambda$  kümesinden yararlanılarak tespit edilmekte ve bu ateşlenme anları da  $BT$  kümesinde sırasıyla verilmektedir.  $T = \{t_1, t_2, t_3\}$  olan bir ZPA ele alalım.  $\tau$  anında sadece  $t_2$  geçişi 3 kez ateşlenebiliyor olsun. Bu durumda  $\tau$  anındaki mevcut  $BT$  kümesi  $BT = \{ [ [ \tau \ \tau + \lambda_{t_2} \ \tau + 2\lambda_{t_2} ] ] \}$  şeklinde verilir. Burada  $t_1$  ve  $t_3$  geçişlerine karşılık boş vektörler vardır.  $ET$  kümesinde ateşlenmesine başlanan geçişlerin ateşlenme tamamlanma anları tutulmaktadır. Bir geçişin ateşlenmesinin tamamlanma anı ateşlenme anına  $D$  kümesinin ilgili geçişe karşılık gelen elemanı ilave edilerek bulunmaktadır. Yine aynı örneğimiz için  $ET$  kümesi  $\tau$  anında  $ET = \{ [ [ \tau + d_{t_2} ] ] \}$  şeklinde oluşmaktadır.  $t_2$  geçişinin ikinci ateşlenmesinin başlama anı olan  $\tau + \lambda_{t_2}$  anında ise bu küme güncellenerek  $ET = \{ [ [ \tau + d_{t_2} \ \tau + \lambda_{t_2} + d_{t_2} ] ] \}$  olarak elde edilir.  $t_2$  geçişinin üçüncü ateşlenmesinin başlama anı olan  $\tau + 2\lambda_{t_2}$  anında ise  $ET = \{ [ [ \tau + d_{t_2} \ \tau + \lambda_{t_2} + d_{t_2} \ \tau + 2\lambda_{t_2} + d_{t_2} ] ] \}$  olarak güncellenmektedir.  $HT$  kümesinde bir ateşlenme tamamlandığında ateşlenebilir hale geçen geçişlerin başlatılan yeni ateşlenmelerinin tamamlanma anları tutulmaktadır. Bu anda  $ET$  kümesinde içinde bulunulan zamandan daha büyük bir elemanın bulunması yani ZPA'da devam etmekte olan bir ateşlenmenin mevcut olması ihtimali vardır. Bu nedenle elde edilen  $HT$  kümesi kullanılarak  $ET$  kümesi güncellenir ve analize devam edilir.

Oluşturulan algoritmalarda sistemler için analiz birim zaman (bz.) kulllanılarak gerçekleştirilmektedir. Bu birim zaman ele alınan sisteme göre saniye, dakika, saat, gün v.b. olabilmektedir. Ana algoritmada kullanıcı tarafından analizin sona erme anı girildi olarak alınmaktadır. Bu sayede örneğin 0 - 50 birim zaman arasındaki ulaşılabilirlik kümesi elde edilmektedir. İçinde bulunulan zaman algoritmalar içerisinde mevcut olan döngüler

kullanılarak adımlar halinde artırılmaktadır. Geçişlere atanmış olan bekleme sürelerinin ilgili *adım* değerinin tam katı olması gerekmektedir. Aksi takdirde bazı olayların atlanması ihtimali bulunmaktadır. Bu nedenle, *adım* değeri geçişlere atanmış bekleme sürelerinin ortak bölenlerinin en büyüğü olarak belirlenmektedir. Buna göre,  $adım = OBEB(\lambda_{t_1}, \lambda_{t_2}, \lambda_{t_3}, \dots, \lambda_{t_{|T|}})$  olmalıdır. Burada, *OBEB* içindeki değerlerin ortak bölenlerinin en büyüğünü bulan fonksiyon olarak verilmektedir.

*RS* kümesindeki sütunlar ulaşılan farklı işaretlemeleri göstermektedir. *RS* kümesinin birinci elemanı  $\{RS\}_1 = m_0$  ve *RSt* kümesinin birinci elemanı  $\{RSt\}_1 = \tau_0$  olmaktadır. Benzer şekilde  $k \in \mathcal{N}$  olmak üzere  $\{RSt\}_k, \{RS\}_k$  ile gösterilen ulaşılabilirlik kümesinin *k.* elemanının elde edilme anını ifade etmektedir. Eğer bir geçiş ulaşılan bir işaretlemede ateşlenebilir halde ise  $F_t$  kümesinin o işaretlemeyle karşılık gelen sütununda bu geçişin indisi verilmekte, eğer başka bir geçiş de ateşlenebilir halde değil ise bu geçiş için aynı sütununda 0 değeri verilmektedir. Böylece  $\{F_t\}_k$  ile gösterilen  $F_t$  kümesinin *k.* sütununda,  $\{RS\}_k$  ile verilen ulaşılabilirlik kümesinin *k.* elemanı olan işaretlemede ateşlenebilir halde bulunan geçişlerin indisleri bulunmaktadır.

Herhangi bir ZPA'da  $\tau_0 = 0$  bz. olmak üzere,  $\tau_f = 1$  bz. olduğunda elde edilen ulaşılabilirlik kümesi ve  $\tau_f = 3$  bz. olduğunda elde edilen ulaşılabilirlik kümesi birbirinden farklıdır. Örneğin Şekil 2.1'deki ZPA için  $[0, 1]$  bz. aralığında

$$RS = R(G, 1) = \{[2 \ 2 \ 0 \ 0 \ 4]^T, [1 \ 1 \ 0 \ 0 \ 2]^T, [1 \ 0 \ 0 \ 0 \ 1]^T, [0 \ 0 \ 0 \ 0 \ 0]^T\}$$

$$RSt = \{0, 0, 0.1, 0.2\}$$

$$F_t = \{[1 \ 2 \ 0 \ 0]^T, [1 \ 2 \ 0 \ 0]^T, [1 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 0]^T\}$$

olarak elde edilirken,  $[0, 3]$  bz. aralığında ise

$$RS = R(G, 3) = \{[2 \ 2 \ 0 \ 0 \ 4]^T, [1 \ 1 \ 0 \ 0 \ 2]^T, [1 \ 0 \ 0 \ 0 \ 1]^T, [0 \ 0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 1 \ 0]^T, [0 \ 0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 1 \ 0]^T, [0 \ 0 \ 1 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 0 \ 0]^T\}$$

$$RSt = \{0, 0, 0.1, 0.2, 2, 2, 2.1, 2.1, 3, 3\}$$

$$F_t = \{[1 \ 2 \ 0 \ 0]^T, [1 \ 2 \ 0 \ 0]^T, [1 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 4]^T, [0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 0 \ 4]^T, [0 \ 0 \ 0 \ 0]^T, [0 \ 0 \ 3 \ 0]^T, [0 \ 0 \ 0 \ 0]^T\}$$

olarak elde edilmektedir.  $RSt$  kümelerinde ilgili ulaşılabilirlik kümelerindeki işaretleme vektörlerinin bz. cinsinden elde edilme anları bulunmaktadır. Örneğin,  $[1 \ 0 \ 0 \ 0 \ 1]^T$  durumuna 0.1 bz.'da ulaşılmaktadır. Bu andaki  $F_t$  kümesinin ilgili sütununa bakıldığında  $t_1$  geçişinin ateşlenebilir olduğu anlaşılmaktadır.

### Örnek 3.1

Şekil 2.1.'deki ZPA'da  $D = [3 \ 2 \ 1 \ 2]$  bz. ve  $\Lambda = [0.2 \ 0.1 \ 0.3 \ 0.4]$  bz. olarak verilmektedir.  $\tau_0 = 0$  bz. anında  $m_0 = [2 \ 2 \ 0 \ 0 \ 4]^T$  başlangıç işaretlemesinin mevcut olduğu görülmektedir. Buna göre,  $\tau_0 = 0$  bz. anında  $\{RS\}_1 = [2 \ 2 \ 0 \ 0 \ 4]^T$ ,  $\{RSt\}_1 = 0$ ,  $\{F_t\}_1 = [1 \ 2 \ 0 \ 0]^T$  olarak elde edilir. Başlangıç işaretlemesine göre  $m_{t_1}(\tau_0) = 2$  ve  $m_{t_2}(\tau_0) = 2$  olmaktadır. Böylece,  $\tau = 0$  bz. anında  $BT$  kümesi  $BT = \{[0 \ 0.2] \ [0 \ 0.1] \ [ ] \ [ ]\}$  şeklinde oluşmaktadır. Burada  $t_1$  geçişine karşılık gelen eleman olan  $[0 \ 0.2]$ ,  $t_1$  geçişinin 0 bz. ve 0.2 bz. anlarında ateşleneceğini göstermektedir. Benzer şekilde  $t_2$  geçişi de 0 bz. ve 0.1 bz. anlarında ateşleneceği bu geçişe karşılık gelen elemandan anlaşılmaktadır.  $t_3$  ve  $t_4$  geçişlerine karşılık gelen  $[ ]$  elemanları ise bu iki geçişin 0 bz. anında ateşlenebilir halde olmadığını belirtmektedir.  $\tau = 0$  bz. anında  $t_1$  ve  $t_2$  geçişlerinin ilk ateşlenmeleri başlatılmakta ve bu iki geçişin girdi yerlerinden belirtiler eksiltilmektedir. Buna bağlı olarak,  $\{RS\}_2 = [1 \ 1 \ 0 \ 0 \ 2]^T$ ,  $\{RSt\}_2 = 0$ ,  $\{F_t\}_2 = [1 \ 2 \ 0 \ 0]^T$  olarak elde edilmektedir.  $\tau = 0$  bz. anında  $ET$  kümesi  $ET = \{[3] \ [2] \ [ ] \ [ ]\}$  şeklinde ifade edilmektedir.  $t_1$  geçişine karşılık gelen  $[3]$  elemanı bu geçişin ateşlenmesinin 3 bz. anında tamamlanacağını göstermektedir.  $t_2$  geçişine karşılık gelen elemandan bu geçişin ateşlenmesini 2 bz. anında tamamlayacağı anlaşılmaktadır.  $t_3$  ve  $t_4$  geçişlerine karşılık gelen elemanların boş vektör olması henüz bu iki geçişin başlatılan ateşlenmelerinin bulunmadığını göstermektedir. Bundan sonra ilk olarak  $t_2$  geçişinin ikinci ateşlenmesine geçilmektedir.  $\tau = 0.1$  bz. anında ilgili ateşlenme başlatılır ve böylece  $\{RS\}_3 = [1 \ 0 \ 0 \ 0 \ 1]^T$ ,  $\{RSt\}_3 = 0.1$ ,  $\{F_t\}_3 = [1 \ 0 \ 0 \ 0]^T$  olarak elde edilir.  $\tau = 0.1$  bz. anında  $ET$  kümesi  $ET = \{[3] \ [2 \ 2.1] \ [ ] \ [ ]\}$  halini alır. Daha sonra  $t_1$  geçişinin

ikinci ateşlenmesine geçilir.  $\tau = 0.2$  bz. anında bu ateşlenme başlatılır ve  $\{RS\}_4 = [0\ 0\ 0\ 0\ 0]^T$ ,  $\{RSt\}_4 = 0.2$ ,  $\{F_t\}_4 = [0\ 0\ 0\ 0]^T$  olarak elde edilir.  $\tau = 0.2$  bz. anında  $ET$  kümesi  $ET = \{[3\ 3.2]\ [2\ 2.1]\ [ ]\ [ ]\}$  şeklinde ifade edilir.  $\tau = \max(BT) = 0.2$  bz. olduğundan  $BT$  kümesi bu anda  $BT = \phi$  olarak güncellenmektedir.  $et = \min(ET) = 2$  olarak elde edilmektedir.  $\tau = 2$  bz. anında  $t_2$  geçişinin  $\tau = 0$  bz. anında başlatılan ateşlenmesi sonlandırılır ve  $\{RS\}_5 = [0\ 0\ 0\ 1\ 0]^T$ ,  $\{RSt\}_5 = 2$ ,  $\{F_t\}_5 = [0\ 0\ 0\ 4]^T$  olarak elde edilir.  $F_t$  kümesi bu anda  $t_4$  geçişinin ateşlenebilir halde olduğunu göstermektedir.  $t_4$  geçişi sadece 1 kez ateşlenebilir halde olduğundan  $\tau = 2$  bz. anında  $BT$  kümesi  $BT = \{[ ]\ [ ]\ [ ]\ [2]\}$  şeklinde oluşur. Buna bağlı olarak hemen  $t_4$  geçişinin ateşlenmesine başlanır ve  $\{RS\}_6 = [0\ 0\ 0\ 0\ 0]^T$ ,  $\{RSt\}_6 = 2$ ,  $\{F_t\}_6 = [0\ 0\ 0\ 0]^T$  olarak elde edilir.  $\tau = 2$  bz. anında  $HT$  kümesi,  $HT = \{[ ]\ [ ]\ [ ]\ [4]\}$  şeklinde oluşmaktadır. Mevcut  $ET$  kümesi,  $HT$  kümesi kullanılarak  $ET = \{[3\ 3.2]\ [2\ 2.1]\ [ ]\ [4]\}$  olarak güncellenir. İçinde bulunulan zaman olan  $\tau = 2$  bz.  $ET$  kümesi tarafından içerildiğinden  $ET$  kümesi güncelenir ve son olarak  $ET = \{[3\ 3.2]\ [2.1]\ [ ]\ [4]\}$  şeklinde ifade edilmektedir. Buna göre  $et = \min(ET) = 2.1$  olarak elde edilir.  $\tau = \max(BT) = 2$  bz. olduğundan bu anda  $BT$  kümesi  $BT = \phi$  olarak güncellenir.  $\tau = 2.1$  bz. anında  $t_2$  geçişinin  $\tau = 0.1$  bz. anında başlatılan ateşlenmesi sonlandırılır ve  $\{RS\}_7 = [0\ 0\ 0\ 1\ 0]^T$ ,  $\{RSt\}_7 = 2.1$ ,  $\{F_t\}_7 = [0\ 0\ 0\ 4]^T$  olarak elde edilir. Bu anda sadece  $t_4$  geçişi bir kez ateşlenebilir halde bulunduğundan  $BT$  kümesi  $BT = \{[ ]\ [ ]\ [ ]\ [2.1]\}$  şeklinde oluşur. Buna bağlı olarak derhal  $t_4$  geçişinin ateşlenmesine başlanır ve  $\{RS\}_8 = [0\ 0\ 0\ 0\ 0]^T$ ,  $\{RSt\}_8 = 2.1$ ,  $\{F_t\}_8 = [0\ 0\ 0\ 0]^T$  olarak elde edilir.  $\tau = 2.1$  bz. anında  $HT$  kümesi,  $HT = \{[ ]\ [ ]\ [ ]\ [4.1]\}$  şeklinde oluşmaktadır. Mevcut  $ET$  kümesi  $HT$  kümesi kullanılarak güncellenir ve  $ET = \{[3\ 3.2]\ [2.1]\ [ ]\ [4\ 4.1]\}$  olarak elde edilir. İçinde bulunulan zaman olan  $\tau = 2.1$  bz.  $ET$  kümesi tarafından içerildiğinden  $ET$  kümesi güncellenir ve  $ET = \{[3\ 3.2]\ [ ]\ [ ]\ [4\ 4.1]\}$  şeklinde verilir. Buna bağlı olarak  $et = \min(ET) = 3$  olarak elde edilmektedir.  $\tau = \max(BT) = 2.1$  bz. olduğundan  $BT$  kümesi bu anda  $BT = \phi$  şeklinde güncellenir.  $\tau = 3$  bz. anında  $t_1$  geçişinin  $\tau = 0$  bz. anında başlatılan

ateşlenmesi sonlandırılmaktadır. Böylece,  $\{RS\}_9 = [0\ 0\ 1\ 0\ 0]^T$ ,  $\{RSt\}_9 = 3$ ,  $\{F_t\}_9 = [0\ 0\ 3\ 0]^T$  olarak elde edilir.  $\tau = 3$  bz. anında mevcut  $BT$  kümesi  $BT = \{[\ ]\ [\ ]\ [3]\ [\ ]\}$  şeklinde oluşur. Buna bağlı olarak  $t_3$  geçişinin ateşlenmesine anlık olarak başlanır ve  $\{RS\}_{10} = [0\ 0\ 0\ 0\ 0]^T$ ,  $\{RSt\}_{10} = 3$ ,  $\{F_t\}_{10} = [0\ 0\ 0\ 0]^T$  olarak elde edilir.  $HT$  kümesi  $\tau = 3$  bz. anında  $HT = \{[\ ]\ [\ ]\ [6]\ [\ ]\}$  şeklinde verilir. Mevcut  $ET$  kümesi  $HT$  kümesi kullanılarak güncellenir ve  $ET = \{[3\ 3.2]\ [\ ]\ [6]\ [4\ 4.1]\}$  şeklinde elde edilir. İçinde bulunan zaman olan  $\tau = 3$  bz.  $ET$  kümesi tarafından içerilmektedir. Bu nedenle  $ET$  kümesi bir kez daha güncellenir ve  $ET = \{[3.2]\ [\ ]\ [6]\ [4\ 4.1]\}$  şeklinde verilir. Böylece,  $et = \min(ET) = 3.2$  olarak bulunmaktadır. Burada,  $\tau = \max(BT) = 3$  bz. olduğundan  $BT$  kümesi bu anda  $BT = \phi$  şeklinde güncellenir. Böylece, eğer analizimizi 0 - 3 bz. aralığında gerçekleştirdiğimizi varsayarsak  $RS = R(G, 3)$ ,  $RSt$  ve  $F_t$  kümeleri şu şekilde elde edilmektedir:

$$RS = \left\{ \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$RSt = \{0, 0, 0.1, 0.2, 2, 2, 2.1, 2.1, 3, 3\}$$

$$F_t = \left\{ \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 3 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

Örneğin,  $RS$  kümesinin 7. elemanı olan  $[0\ 0\ 0\ 1\ 0]^T$  işaretleme vektörü, 2.1 bz.'da elde edilmiş ve bu anda sadece  $t_4$  geçişi ateşlenebilir.

### Örnek 3.2

$R(G, \tau_f)$ 'nin oluşturulması sırasında, çelişki durumunun olduğu bir örneği ele alalım. Şekil 2.1.'deki ZPA'da  $m_0 = [2\ 2\ 0\ 0\ 2]^T$ ,  $D$  kümesi

$D = [3 \ 2 \ 2 \ 1]^T$  bz. ve  $\Lambda$  kümesi de  $\Lambda = [0.05 \ 0.03 \ 0.02 \ 0.01]^T$  bz. olarak değiştirilmiş ve  $[0, 5]$  bz. aralığında analiz gerçekleştirilmiştir.  $\tau_0$  anında  $t_1$  ve  $t_2$  geçişlerinin ateşlenebilirlik dereceleri,

$$m_{t_1}(\tau_0) = \min(\text{ceil}(M(p_1, \tau_0) / 1), \text{ceil}(M(p_5, \tau_0) / 2)) = \min(2, 1) = 1$$

$$m_{t_2}(\tau_0) = \min(\text{ceil}(M(p_2, \tau_0) / 1), \text{ceil}(M(p_5, \tau_0) / 2)) = \min(2, 1) = 1$$

olarak elde edilmektedir. Buna bağlı olarak ilgili geçişlerin ateşlenmesi ile analize başlanmaktadır.  $[0, 5]$  bz. aralığındaki analiz tamamlandığında  $RS$ ,  $RSt$  ve  $F_t$  kümeleri şu şekilde elde edilmektedir:

$$RS = \left\{ \begin{array}{l} \left[ \begin{array}{c} 2 \\ 2 \\ 0 \\ 0 \\ 2 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 1 \\ 0 \\ 1 \end{array} \right], \left[ \begin{array}{c} 0 \\ 2 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 0 \\ 0 \\ 1 \end{array} \right], \left[ \begin{array}{c} 0 \\ 2 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 2 \\ 1 \\ 0 \\ 1 \\ 1 \end{array} \right], \\ \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \end{array} \right\}$$

$$RSt = \{0, 0, 2, 2, 3, 3, 5, 5, 3, 5, 5, 5\}$$

$$F_t = \left\{ \begin{array}{l} \left[ \begin{array}{c} 1 \\ 2 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 4 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 3 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 0 \\ 4 \end{array} \right], \\ \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \end{array} \right\}$$

Bu kümeler incelendiğinde çeşitli anlarda çelişki durumunun olduğu ve bu durumda seçilerek ateşlenen geçişlere bağlı olarak farklı işaretlemelerin



elde edildiđi görülmüştür. Örneđin  $RS$  kümesinin 5. elemanı olan işaretlemede  $t_1$ ,  $t_2$  ve  $t_3$  geçişleri ateşlenebilirlik olarak görünmektedir. Ancak  $p_5$  yerindeki beirti sayısına bađlı olarak  $t_1$  ve  $t_2$  geçişleri aynı anda ateşlenememektedir. Bu işaretlemeden sonra  $t_1$  ve  $t_3$  geçişleri seçilerek ateşlendiđinde  $RS$  kümesinin 6. elemanı,  $t_2$  ve  $t_3$  geçişleri seçilerek ateşlendiđinde ise  $RS$  kümesinin 9. elemanı elde edilmiştir. Oluşan diđer çelişki durumlarında da benzer yaklaşım sergilenmiş ve bu sayede elde edilebilir bütün işaretlemelemlerin ulaşılabilirlik kümesinde yer alması sağlanmıştır.

#### 4. SİSTEM ÇIKMAZI

Ulaşılabilirlik kümesi elde edildikten sonra sistem çıkmazının oluşup oluşmadığının analizine geçilmektedir.

**Tanım 4.1:** Eğer ulaşılan  $\mathcal{M} \in R(G, \tau_f)$  işaretleme vektöründe, ateşlenebilirlik koşulunu sağlayan geçiş bulunmamakta ( $E(\mathcal{M}) = \emptyset$ ) ve bu anda devam etmekte olan bir ateşlenme bulunmamakta ise, ZPA'da sistem çıkmazı oluşmaktadır.

Ulaşılabilirlik kümesi iki farklı işlem sonucunda elde edilmiş olan işaretlemeleri içermektedir. Bunlardan ilki ateşlenmelerin başlatılıp belirtilerin eksiltilmesi işlemi diğeri ise ateşlenmelerin tamamlanıp yeni belirtilerin çıktığı yerlerinde oluşturulması işlemidir. İlk işlem sonucu meydana gelen işaretlemeler ara veya geçici işaretlemeler olarak adlandırılabilir. Bunun nedeni ateşlenmelerin sadece başlatılmış olması ve bu ateşlenmelerin sonucunda çıktığı yerlerinde belirtilerin oluşturulup Petri ağının çalışmasına devam edebileceğidir. Bu nedenle, sistem çıkmazının tespiti algoritmasında, sadece ateşlenmelerin tamamlanma anlarında elde edilmiş olan işaretlemeler ele alınmaktadır.

ZPA'da sistem çıkmazının oluşup oluşmadığının tespiti daha detaylı olarak şu şekilde verilebilir:

$\mathcal{M} \in R(G, \tau_f)$ ,  $\tau$  anındaki işaretleme vektörünü göstermek üzere,

- 1) Eğer  $E(\mathcal{M}) \neq \emptyset$  ise, ulaşılan işaretlemelerde ateşlenebilir olan geçişler mevcuttur ve bunların ateşlenmelerine geçilir. Bu durumda ZPA çalışmasına devam etmektedir ve sistem çıkmazı meydana gelmemiştir.
- 2) Eğer  $E(\mathcal{M}) = \emptyset$  ( $\{RS\}_k = \mathcal{M}$  olmak üzere,  $\{F_t\}_k = [0 \dots 0]^T$  olmaktadır,  $k \in \mathcal{N}$ ) ise, ZPA'da ulaşılan işaretlemelerde ateşlenebilir olan geçişler bulunmamaktadır. Bu durumda;

2-a) Eğer  $\tau = \max(ET)$  ise, ZPA'da sistem çıkmazı oluşmaktadır.

2-b) Eğer  $\tau < \max(ET)$  ise, ZPA'da sistem çıkmazı oluşmamaktadır. Çünkü bu durumda devam eden ateşlenmeler bulunmaktadır ve ZPA'nın çalışması devam etmektedir. Bundan sonra  $ET$  kümesinin maksimum elemanına ulaşıldığında yeniden bir kontrol yapılması gerekmektedir.

Ek-D'de verilen Sistem Çıkmazı algoritmasında sistem çıkmazının olduğu işaretlemeler tespit edilmekte ve  $DM$  kümesinde saklanmaktadır. Sistem Çıkmazı algoritması için  $G$  ile gösterilen ZPA'yı tanımlayan bilgilerin,  $ET$  kümesinin, zaman artırımını gösteren  $adım$  değerinin ve analizin sona erme zamanının girdi olarak girilmesi gerekmektedir.  $DMt$  kümesinde ise sistem çıkmazının olduğu işaretlemelerin hangi anda meydana geldiği bilgisi tutulmaktadır.  $k \in \mathcal{N}$  olmak üzere,  $DMt(k)$  elemanı bize  $DM$  kümesinin  $k$ . elemanı olan işaretlemenin hangi anda elde edildiğini ifade etmektedir. Eğer  $DM$  kümesi belirtilen aralıkta gerçekleştirilen analiz sonucunda başlangıçta olduğu gibi boş küme olarak kalmakta ise bu aralıkta ele alınan ZPA'da sistem çıkmazının meydana gelmediği bir mesaj ile kullanıcıya bildirilmektedir.

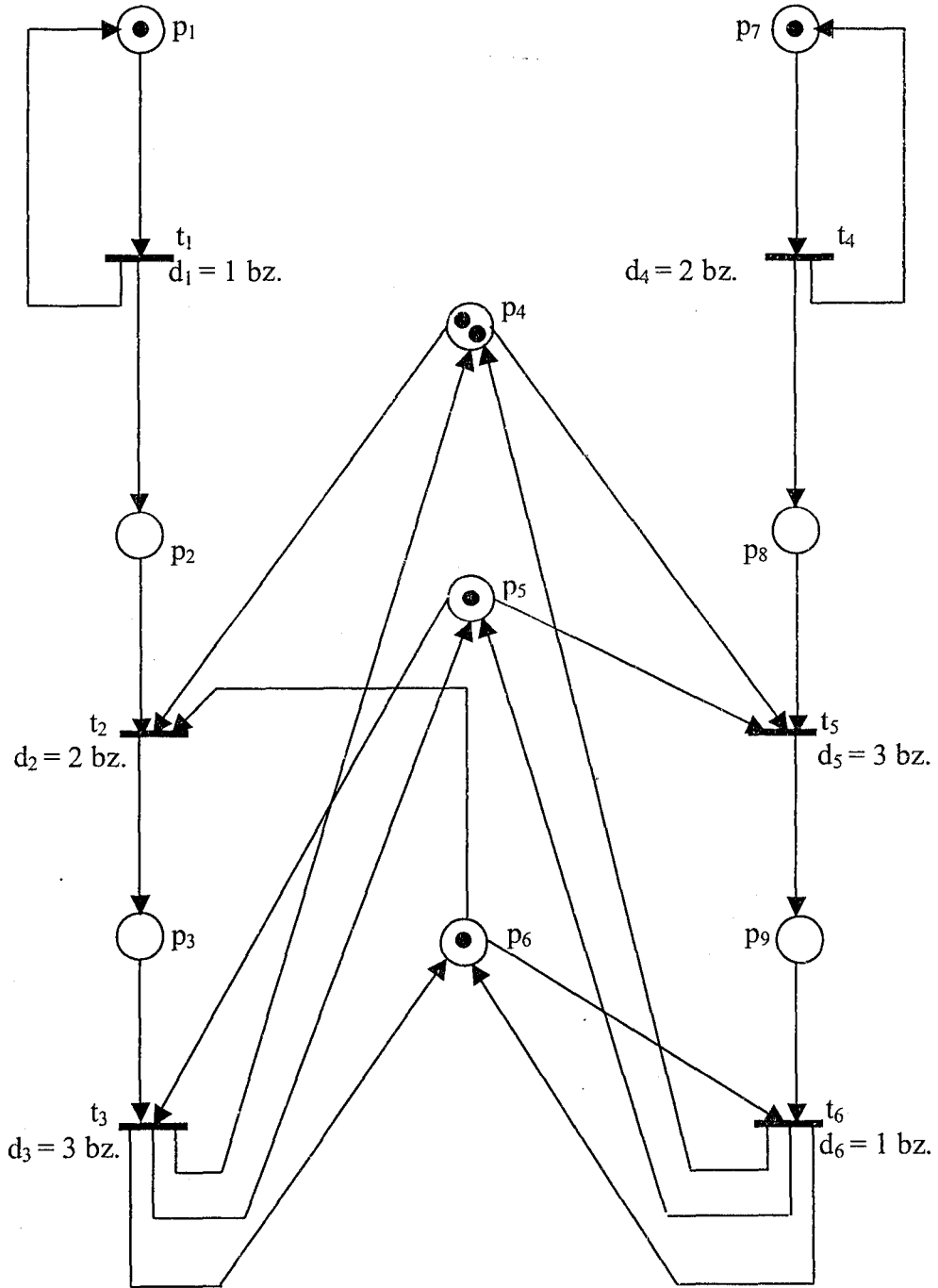
## 5. UYGULAMALAR

Bu bölümde oluşturulan MATLAB programının farklı ZPA'lar üzerine uygulanmasıyla elde edilen sonuçlar verilecektir. Geliştirilen algoritmalara uygun olarak oluşturulan MATLAB programına farklı ZPA'ların yapısal özellikleri girdi olarak verilmiş ve çeşitli aralıklarda analiz gerçekleştirilmiştir. Analizi gerçekleştirilen ZPA'larda  $\Lambda$  kümesindeki değerlerin ortak bölenlerinin en büyüğü 0.01 olarak belirlenmiş ve zaman artırımları 0.01 bz.'lık adımlarla gerçekleştirilmiştir.

### Örnek 5.1

Şekil 5.1.'deki ZPA için başlangıç işaretlemesi  $m_0 = [1 \ 0 \ 0 \ 2 \ 1 \ 1 \ 1 \ 0 \ 0]^T$ , geçişlere atanmış olan ateşlenme gecikmelerinin kümesi  $D = [1 \ 2 \ 3 \ 2 \ 3 \ 1]$  bz. ve geçişlere ait bekleme sürelerinin kümesi  $\Lambda = [0.03 \ 0.02 \ 0.04 \ 0.01 \ 0.02 \ 0.05]$  bz. olarak verilmektedir.  $\tau_0 = 0$  bz. ve  $\tau_f = 10$  bz. olarak alınmıştır. Elde edilen  $RS$ ,  $RSt$  ve  $F_t$  kümeleri ( $|RS| = |RSt| = |F_t| = 22$ ) EK-G'de verilmiştir.

Şekil 5.1.'deki ZPA sınırsız bir Petri ağıdır ( $p_2$  ve  $p_8$  yerlerindeki belirtiler  $t_1$  ve  $t_2$  geçişleri ile sürekli sürekli artabilmektedir) .  $\tau_f = 10$  bz. olarak belirlenmesiyle 0-10 bz. arasındaki olası tüm işaretleme vektörleri elde edildiğinden, Petri ağının sınırsız olması önemli olmamaktadır. Eğer istenirse, bir sınır vektörü de baştan belirlenerek,  $\tau_f$  süresine bağımlı olmaksızın ulaşılabilirlik kümesi elde edilebilmektedir. Bahsi geçen sınır vektörü,  $K = [k_1 \ k_2 \ \dots \ k_{|P|}]$  ( $k_i \in \mathcal{N}$ ,  $i \in \{1, \dots, |P|\}$ ) şeklinde olup  $\forall M \in RS$  işaretleme vektörü için  $M(i) \leq K(i)$ ,  $\forall i \in \{1, \dots, |P|\}$  şartı sağlanmalıdır. Örneğimiz için sınır vektörü,  $K = [9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9 \ 9]^T$  olarak alınmıştır. Ancak ilgili ZPA'da bir yerde bulunabilecek maksimum belirti olan 9 belirtiye en erken 10 bz. anında ulaşıldığından girilen aralıkta  $K$  vektörünün bu şekilde belirlenmesi ulaşılabilirlik kümesi için sınırlayıcı olmamaktadır. Bunlara bağlı olarak ilgili ZPA'da  $[0, 10]$  bz. aralığında analiz gerçekleştirilmiş ve  $RS$ ,  $RSt$  ve  $F_t$  kümeleri elde edilmiştir. Gerçekleştirilen analiz sonucunda ZPA'da sistem çıkmazının oluşmadığı bir mesaj ile kullanıcıya bildirilmiştir.



Şekil 5.1. Örnek 5.1'de ele alınan ZPA

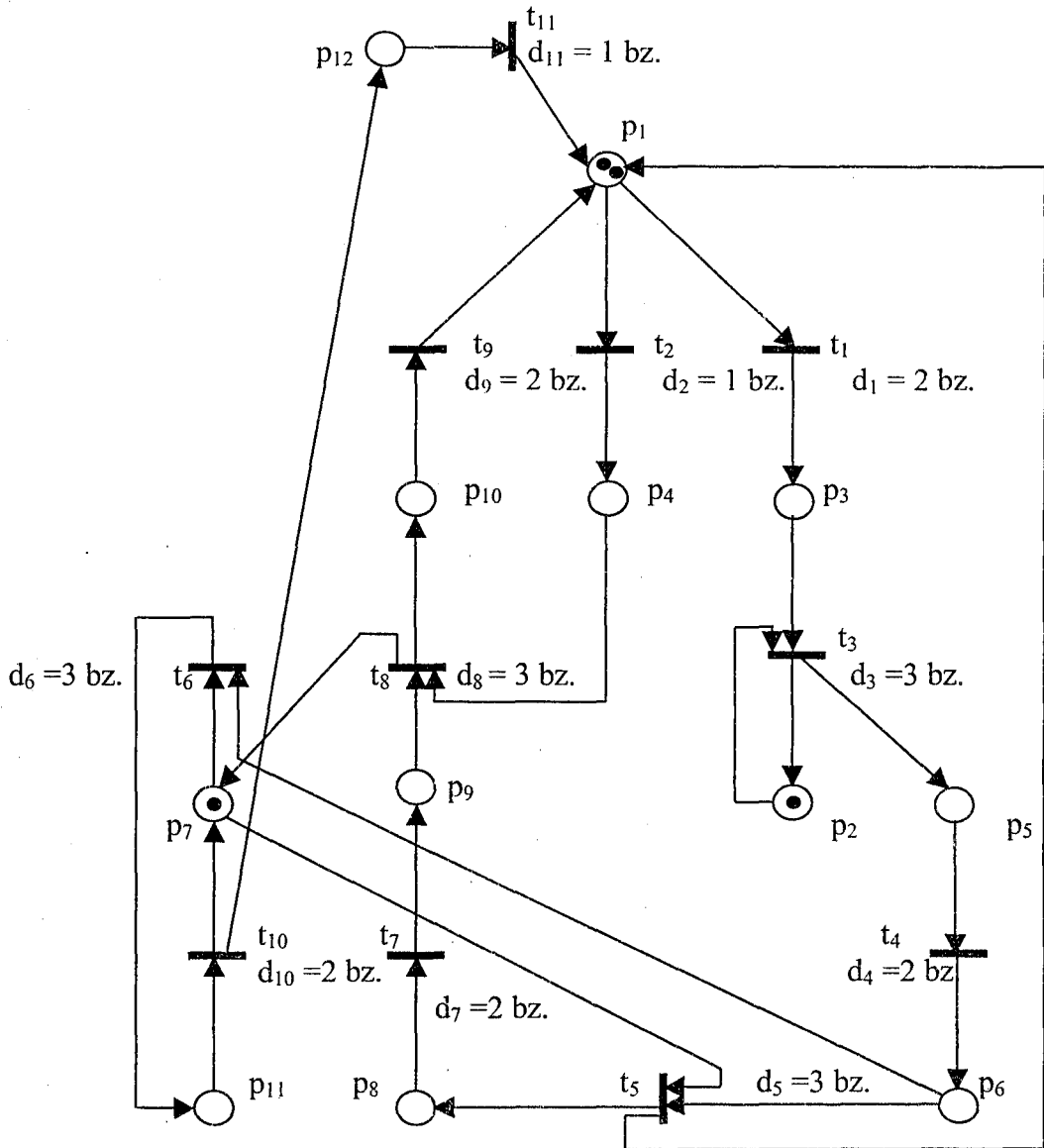
Eğer örneğimiz için sınır vektörü  $K = [9\ 9\ 9\ 9\ 9\ 9\ 9\ 9\ 9]^T$  olarak tutulup  $\tau_f$  değeri artırılırsa ZPA'nın ulaşılabilirlik kümesinin eleman sayısının artmadığı görülmüştür. Eğer bir yerde bulunabilecek maksimum belirti sayısı 15 olarak belirlenir ( $K = [15\ 15\ 15\ 15\ 15\ 15\ 15\ 15\ 15]^T$ ) ve  $\tau_f = 100$  bz. olarak alınır, Şekil 5.1.'deki örnek ZPA için  $|RS| = |RSt| = |F_t| = 34$  olarak elde edilmektedir.

### Örnek 5.2

Şekil 5.2.'deki ZPA için  $\tau_0 = 0$  bz. anındaki başlangıç işaretlemesi  $m_0 = [2\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$ , geçişlere atanmış olan ateşlenme gecikmelerinin kümesi  $D = [2\ 1\ 3\ 2\ 3\ 3\ 2\ 3\ 2\ 2\ 1]$  bz. ve geçişlere ait bekleme sürelerinin kümesi  $\Lambda = [0.02\ 0.03\ 0.01\ 0.04\ 0.03\ 0.05\ 0.02\ 0.02\ 0.04\ 0.05\ 0.02]$  bz. olarak verilmektedir.  $\tau_f = 20$  bz. olarak alınmıştır. Analiz sonucunda elde edilen  $RS$ ,  $RSt$  ve  $F_t$  kümeleri ( $|RS| = |RSt| = |F_t| = 46$ ) EK-G'de verilmiştir.

EK-G'de verilen  $RSt$  kümesindeki değerlere dikkat edilirse ZPA'da ilk önce başlangıç anı olan 0 bz.'dan başlayarak  $\tau_f = 20$  bz. anına kadar değerlerin elde edildiği görülmektedir. Ardından oluşan çeşitli çelişki durumları için çelişkinin oluşma anından başlayarak yine 20 bz.'a kadar olan değerler elde edilmiştir. Bunun nedeni çelişkinin oluştuğu anda ateşlenebilirlik koşulunu sağlayan birden fazla geçişten sadece bir tanesinin ateşleniyor olmasıdır. Sadece 11 bz.'da oluşan çelişki durumundan sonra 19 bz.'a kadar olan değerler elde edilmiştir. Bunun nedeni, 19 bz. anında başlatılan ateşlenmenin  $\tau_f = 20$  bz.'dan daha geç bir değerde sonlandırılacak olmasıdır. En sonda elde edilen 14 ve 18 bz. değerleri ise sistem çıkmazının oluştuğu işaretlemelerin elde edilme anlarıdır ve bunlardan sonra ilgili ZPA'da yeni bir işaretleme vektörü elde edilmemektedir.

Bunların yanında  $[0, 20]$  bz. aralığı için gerçekleştirilen analiz sonucu Şekil 5.2.'deki ZPA'da sistem çıkmazının oluştuğu iki adet işaretlemeye ulaşılmıştır. Sırasıyla bu işaretlemeleri ve elde edilme anlarını gösteren  $DM$  ve  $DMt$  kümeleri de EK-G'de verilmektedir ( $|DM| = |DMt| = 2$ ).



Şekil 5.2. Örnek 5.2'de ele alınan ZPA

14 bz. anında elde edilen sistem çıkmazının olduğu işaretleme vektörüne  $t_1, t_2, t_3, t_4, t_6, t_{10}, t_{11}, t_2$  ateşlenme dizisindeki geçişlerin ateşlenmesi sonucunda ulaşılmaktadır. 18 bz. anında elde edilen sistem çıkmazının meydana geldiği işaretleme vektörüne ise  $t_1, t_2, t_3, t_4, t_5, t_7, t_8, t_9, t_2$  ateşlenme dizisindeki geçişlerin ateşlenmesi sonucunda ulaşılmaktadır.

Aynı ZPA için  $[0, 100]$  bz. aralığında gerçekleştirilen analiz sonucu incelendiğinde  $|RS| = |RSt| = |F_t| = 2361$  olduğu görülmüştür. Ayrıca ZPA'nın  $[0, 100]$  bz. aralığındaki analizinde sistem çıkmazının olduğu farklı zamanlarda elde edilen toplam 76 adet işaretleme vektörü bulunduğu gözlenmiştir.

Şekil 5.2.'deki ZPA'da geçişlere atanmış olan ateşlenme gecikmelerinin kümesi  $D = [1\ 2\ 2\ 1\ 2\ 2\ 1\ 2\ 3\ 1\ 2]$  bz. olarak değiştirilmiş ve  $[0, 20]$  bz. aralığında gerçekleştirilen analiz sonucunda elde edilen  $DM$  ve  $DMt$  kümeleri EK-G'de verilmiştir ( $|DM| = |DMt| = 4$ ).

Ele alınan örneklerde  $\tau_0 = 0$  olarak alınmıştır.  $\tau_0 \neq 0$  olduğu durumlarda ise başlangıç işaretlemesindeki belirtilerin ZPA'daki yerlere bu anda bulunduğu kabul edilmekte ve analiz  $\tau_0$  ile belirtilen andan itibaren gerçekleştirilmektedir. Buna göre  $a, b \in \mathcal{N}$  olmak üzere; bir ZPA'da  $[0, a]$  bz. aralığında elde edilen ulaşılabilirlik kümesinin eleman sayısı ile  $[b, a + b]$  bz. aralığında elde edilen ulaşılabilirlik kümesinin eleman sayısı aynı olmaktadır. Sadece bu iki ulaşılabilirlik kümesinde içerilen işaretleme vektörlerinin elde edilme zamanları yani bu kümelere karşılık gelen  $RSt$  kümelerinin elemanları farklılık göstermektedir.



## 6. SONUÇLAR

Bu tez çalışmasında, Petri ağında bulunan geçişlere belirli ateşlenme gecikmeleri atanarak ZPA elde edilmiştir. Ele alınan ZPA'da aynı anda veya devam eden ateşlenmeler bulunmakta iken birden fazla geçişin ateşlenebilmesine olanak sağlayan bir matematiksel model sunulmuştur.

Ele alınan model için bir ZPA'da belirli bir zaman aralığında ulaşılabilirlik kümesi elde edilmesine yönelik algoritmalar oluşturulmuştur. Elde edilen ulaşılabilirlik kümesinde bulunan her bir işaretleme vektörüne bz. cinsinden hangi anda ulaşıldığı da tespit edilmiştir. Ayrıca ulaşılan her bir işaretlemede hangi geçişlerin ateşlenebilir olduğu da bulunmuştur. Ardından elde edilen bu değerler kullanılarak, sistem çıkmazı analizi gerçekleştirilmiş ve sistem çıkmazına yol açan işaretleme vektörleri ile bu işaretleme vektörlerinin elde edilme anları bulunmuştur. Mevcut sonuçlar  $f_t = e_t$  alınarak elde edilmiştir. Eğer  $f_t \geq e_t$  olarak alınırsa, sunulan algoritmaların değiştirilmesi gerekmektedir (geçişlerin ateşlenme zamanları,  $f_t, \forall t \in T$ , da bilinmelidir).

ZPA'nın ulaşılabilirlik kümesinde zaman ifadelerinin eklenmesinden dolayı ara veya geçici işaretlemelemlerin de bulunduğu ve bu yüzden bu kümenin aynı Petri ağının zamanlandırılmamış halinin ulaşılabilirlik kümesine göre daha geniş bir küme olduğu görülmüştür.

Oluşturulan algoritmalar EK-F'de verilen MATLAB programı kullanılarak gerçekleştirilmiştir. Buradaki birim zaman sistemin yapısına göre saniye, dakika, saat, v.b. olabilmektedir. Analiz bz. cinsinden belirli bir aralıkta gerçekleştirilmesine rağmen, eğer istenirse analizin ne kadar sürdüğü saniye cinsinden ölçülebilir. Farklı ZPA'lar için gerçekleştirilen analizlerin farklı sürelerde tamamlandığı görülmüştür. Örneğin, Şekil 5.1.'deki ZPA için  $[0, 10]$  bz. aralığındaki analiz 1.2030 saniye sürmüştür ve analiz sonucunda  $|RS| = 22$  olduğu görülmüştür. Programlar, Pentium 4, 1.60 GHz CPU, 384 MB RAM'e sahip bir kişisel bilgisayarda çalıştırılmıştır. Daha sonra 17 yer ve 11 geçişe sahip başka bir ZPA için ise  $[0, 20]$  bz. aralığında analiz gerçekleştirilmiştir. Bu analizin 1333.9 saniye sürdüğü ve analiz sonucunda  $|RS| = 2516$  olduğu görülmüştür. Aybar ve İftar tarafından sunulan

çalışmada, [Aybar ve İftar, 2003], Petri ağındaki yerlerin, geçişlerin ve bunlar arasındaki bağlantıların sayısına bağlı olarak işlem karmaşıklığının arttığı ve işlem süresinin buna bağlı olarak belirlendiğine değinilmiştir. Aynı olay ZPA'lar için de söz konusudur. Ayrıca, girilen zaman aralığının daha uzun seçilmesi, olası işaretleme vektör sayılarını arttırdığından ZPA'larda gerçekleştirilen analizin tamamlanma süresi artmaktadır. Bunun yanında, ZPA'da ulaşılan işaretlemelerde oluşan çelişki durumlarının sayısının artması analizlerin daha da karmaşıklaşmasına ve sürelerinin uzamasına neden olmaktadır. ZPA'lar için de örtüşmeli ayrıştırma ve genişletme yöntemleri kullanılarak analiz süresinin daha da kısaltılabileceği düşünülmektedir.

ZPA'lar için ulaşılabilirlik kümesi elde edildikten sonra, Petri ağının farklı özellikleri (canlılık, tersine dönüşebilirlik v.b.) incelenebilir. Gelecekteki çalışmalarda, bu çalışmadakine benzer analizler yerler ve geçişler arasındaki bağlantıların ağırlık değerlerinin birden farklı olarak verildiği ZPA'lar için de gerçekleştirilebilir.

## KAYNAKLAR

AYBAR, A., *Petri ağlarında örtüşmeli ayrıştırma ve geliştirme kullanılarak kontrolör tasarımı*, Doktora Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, Türkiye, (2001).

AYBAR, A. ve İFTAR, A., *Decentralized supervisory controller design to avoid deadlock in Petri nets*, Int. J. Control, **76**, 13, 1285-1295, (2003).

BARAD, M., *Timed Petri nets as a verification tool*, Proceedings of the 1998 Winter Simulation Conference, (Ed: MEDEIROS, D.J., WATSON, E.F., CARSON, J.S., MANIVANNAN, M.S.), Washington, USA, 547-554, (1998).

BERTHOMIEU, B. ve MENASCHE, M., *An enumerative approach for analyzing time Petri nets*, Proceedings of IFIP Cong., Paris, 41-46, (1983).

BOWDEN, F.D.J., *A brief survey and synthesis of the roles of time in Petri nets*, Mathematical and Computer Modelling, **31**, 55-68, (2000).

BUY, U. ve SLOAN, R.H., *Analysis of real-time programs with simple time Petri nets*, ISSTA '94. Proceedings of the 1994 International Symposium on Software Testing and Analysis, ACM Press, Seattle, 228-239, (1994).

CHIOLA G. ve FERSCHA, A., *Exploiting timed Petri net properties for distributed simulation partitioning*, Proceedings of the 26th Int. Conf. on System Sciences, IEEE Computer Society Press, Washington, USA, 194-203, (1993).

COOLAHAN, J.E.Jr. ve ROUSSOPOULOS, N., *Timing requirements for time driven systems using augmented Petri nets*, IEEE Transactions on Software Engineering, **SE-9**, 603-616, (1983). [Tsai ve ark., 1995].

DESROCHERS, A.A. ve AL-JAAR, R.Y., *Applications of Petri nets in manufacturing systems*, The Institute of Electrical and Electronics Engineers, Inc., New York, USA, (1995).

FREEDMAN, P., *Time, Petri nets, and robotics*, IEEE Transactions on Robotics and Automation, **7**, 4, 417-433, (1991).

GAUBERT, S. ve MAIRESSE, J., *Modeling and analysis of timed Petri nets using heaps of pieces*, IEEE Transactions on Automatic Control, **44**, 4, 683-697, (1999).

HO, Y.(Ed.), *Discrete event dynamic systems: analyzing complexity and performance in the modern world*, The Institute of Electrical and Electronics Engineers, Inc., New York, USA, (1992).

HUIFANG, L. ve YUSHUN, F., *Schedulability analysis method of timing constraint Petri nets*, Tsinghua Science and Technology, **7**, 6, 596-601, (2002).

JANNECK, J.W., *Behavioural prediction of time Petri nets with applications to distributed simulation*, Proceedings HPC '98, High Performance Computing, Boston, MA, 416-424, (1998).

KOBAYASHI, K., INOUE, K. ve USHIO, T., *LLP supervisory control with timed Petri net models in mobile robots*, Proceedings of the 2001 International Conference on Systems, Man and Cybernetics, Arizona, USA, 3229-3234, (2001).

KOUTSOUKOS, X.D., ANTSAKLIS, P.J., HE K.X. ve LEMMON, M.D., *Programmable timed Petri nets in the analysis and design of hybrid control systems*, Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, FL, 1617-1622, (1998).

LEFEBVRE, D. ve EL MOUDNI, A., *Firing and enabling sequences estimation for timed Petri nets*, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, **31**, 3, 153-162, (2001).

LILIUS, J., *Efficient state space search for time Petri nets*, Proceedings of MFCS Workshop on Concurrency, Brno '98, Electronic Notes in Theoretical Computer Science, **18**, Elsevier, (1998).

LING, S. ve SCHMIDT, H.W., *A notion of safeness in time for Petri nets*, Technical Report TR96/21, Department of Software Development, Monash University, 1-13, (1996).

LÓPEZ-MELLADO, E., *Analysis of discrete event systems by simulation of timed Petri net models*, Mathematics and Computers in Simulation, **61**, 53-59, (2002).

MERLIN, P., *A study of the recoverability of computer systems*, Doktora Tezi, University of California, Irvine, California, USA, (1974). [Berthomieu ve Menasche, 1983].

PETRI, C.A., *Kommunikation mit automaten*, Doktora Tezi, University of Bonn, West Germany, (1962). [Bowden, 2000].

RAMCHANDANI, C., *Analysis of asynchronous concurrent systems by timed Petri nets*, Doktora Tezi (MAC Technical Report-120), L.C.S. M.I.T., Cambridge, Massachusetts, (1974).

SIFAKIS, J., *Use of Petri nets for performance evaluation*, Measuring, Modelling and Evaluating Computer Systems, (Ed: BEILNER, H. ve GELLENBE, E.), Kuzey Hollanda, 75-93, (1977). [Bowden, 2000].

TSAI, J.J.P., YANG, S.J. ve CHANG, Y.-H., *Timing constraint Petri nets and their application to schedulability analysis of real-time system specifications*, IEEE Transactions on Software Engineering, **21**, 1, 32-49, (1995).

WANG, J., *Timed Petri nets: theory and application*, Kluwer Academic Publishers, Boston, (1998).

ZHOU, M.C. ve DICESARE, F., *Petri net synthesis for discrete event control of manufacturing systems*, Kluwer Academic Publishers, Boston, (1993).

ZUBEREK, W.M., *Timed Petri nets: definitions, properties, and applications*, Microelectron. Reliab., **31**, 4, 627-644, (1991).

ZUBEREK, W.M., *Timed Petri nets in modelling and analysis of cluster tools*, IEEE Transactions on Robotics and Automation, **17**, 5, 562-575, (2001).

## EKLER

Burada verilen algoritmalar pseudo-code ile yazılmıştır.

### A) Ana Algoritma

$G$  Petri ağına ait bilgileri yükle

Analizin bitiş zamanını ( $\tau_f$ ) gir

UK-1 algoritmasını çağır

UK-2 algoritmasını çağır

Sistem Çıkmazı algoritmasını çağır

BİTİR

### B) $m_0$ işaretlemesinde ateşlenebilir halde olan geçişlerin ateşlenmelerinin başlatıldığı UK-1 algoritması

UK-1[ $G$ ]

$F_t = \{E(m_0)\}$ ,  $RS = \{m_0\}$ ,  $RSt = \{\tau_0\}$

$BT = \phi$

For  $j = 1$  To  $|T|$

If  $T(j) \in E(m_0)$  Then

For  $u = 1$  To  $m_{T(j)}(\tau_0)$

$\{BT\}_j(u) = \tau_0 + (u - 1) \lambda_{T(j)}$

End

End

End

$\tau_1 = \max(BT)$

$Q = OBE B_{\forall t \in T}(\lambda_t)$

$k = 1$ ,  $ET = \phi$

$Z = [ ]$

For  $j = 1$  to  $|T|$

$z = \text{length}(\{BT\}_j)$

$Z \leftarrow Z \hat{\cup} z$

End

For  $\tau = \tau_0$  to  $\tau_1$  step  $Q$

For  $j = 1$  to  $|T|$

If  $\text{length}(\{BT\}_j) \geq k$  Then

If  $\{BT\}_j(k) == \tau$  Then

$\mathcal{M}' = \rho_I(\mathcal{M}, T(j))$

$\mathcal{M} = \mathcal{M}'$

$\{ET\}_j(k) = \{BT\}_j(k) + D(j)$

End

End

End

$RS \leftarrow RS \cup \{\mathcal{M}\}$

$RSt \leftarrow RSt \cup \{\tau\}$

$F_t \leftarrow F_t \cup \{E(\mathcal{M})\}$

If  $k < mx(Z)$  Then

$k \leftarrow k + 1$

End

If  $\tau == \tau_1$  Then

$BT = \phi$

End

$et = \min(ET)$

End

C) Başlatılan ateşlenmelerin sıra ile tamamlandığı, daha sonra  $\tau_f$  anına kadar ateşlenmelerin tekrar başlatılıp sonlandırıldığı ve ulaşılabilirlik kümesinin son halinin elde edildiği UK-2 algoritması

UK-2[ $G, ET, BT, \tau_f, RS, RSt, Ft, \tau_1, Q, M(\tau_1), et$ ]

Do Loop UK-Final

For  $\tau = \tau_1$  to  $et$  step  $Q$

For  $j = 1$  to  $|T|$

If  $\{ET\}_j \neq [ ]$  Then

If  $\{ET\}_j(1) == \tau$  Then

$\mathcal{M}'' = \rho_O(\mathcal{M}, T(j))$

$\mathcal{M} = \mathcal{M}''$

End

End

End

For  $i = 1$  to  $|RS|$

If  $\{RS\}_i == \mathcal{M}$  and  $\{RSt\}_i == \tau$  Then

$kk = 0$

Go To Break 1

Else

$kk = 1$

End

End

Break 1: Continue

If  $kk == 1$  Then

$RS \leftarrow RS \cup \{\mathcal{M}\}$



$$RSt \leftarrow RSt \cup \{\tau\}$$

$$F_t \leftarrow F_t \cup \{E(\mathcal{M})\}$$

End

End

$$k = 1, Q = 0.01, HT = \phi$$

$$\tau_1 = \max(BT)$$

$$Z = [ ]$$

For  $j = 1$  to  $|T|$

$$z = \text{length}(\{BT\}_j)$$

$$Z \leftarrow Z \hat{\cup} z$$

End

For  $\tau = et$  to  $\tau_1$  step  $Q$

For  $j = 1$  to  $|T|$

If  $\text{length}(\{BT\}_j) \geq k$  Then

If  $\{BT\}_j(k) == \tau$  Then

$$\mathcal{M}' = \rho_T(\mathcal{M}, T(j))$$

$$\mathcal{M} = \mathcal{M}'$$

$$\{HT\}_j(k) = \{BT\}_j(k) + D(j)$$

$$ET \leftarrow ET \cup HT$$

End

End

End

For  $i = 1$  to  $|RS|$

If  $\{RS\}_i == \mathcal{M}$  and  $\{RSt\}_i == \tau$  Then

$$kk = 0$$

Go To Break 2

```

Else
     $kk = 1$ 
End
End
End
Break 2: Continue
If  $kk == 1$  Then
     $RS \leftarrow RS \cup \{\mathcal{M}\}$ 
     $RSt \leftarrow RSt \cup \{\tau\}$ 
     $F_t \leftarrow F_t \cup \{E(\mathcal{M})\}$ 
End
If  $k < mx(Z)$  Then
     $k \leftarrow k + 1$ 
End
If  $\tau == \tau_1$  Then
     $BT = \phi$ 
End
For  $j = 1$  to  $|T|$ 
    If  $length(\{ET\}_j) \geq 1$  Then
        If  $\{ET\}_j(1) == \tau$  Then
            If  $length(\{ET\}_j) == 1$  Then
                 $\{ET\}_j = [ ]$ 
            Else
                 $B = [ ], L = (length(\{ET\}_j) - 1)$ 
                For  $u = 1$  to  $L$ 
                     $B(u) = \{ET\}_j(u + 1)$ 
                End
            End
        End
    End
End

```

```

                                {ET}_j = B
                                End
                                End
                                End
                                End
                                et = min(ET)
                                End
                                If  $\tau \geq \tau_f$  Then
                                    Exit Loop UK-Final
                                End
                                Loop UK-Final

```

D) ZPA'da sistem çıkmazının meydana gelip gelmediğini test eden

Sistem Çıkmazı algoritması

Sistem Çıkmazı[ $G, ET, Q, \tau_f$ ]

$DM = \phi, DMt = \phi$

For  $\tau = \tau_0$  to  $\tau_f$  step  $Q$

If  $E(\mathcal{M}) == \emptyset$  Then

If  $\max(ET) \leq \tau$  Then

If  $DM \neq \phi$  Then

For  $i = 1$  to  $|DM|$

If  $\{DM\}_i == \mathcal{M}$  and  $\{DMt\}_i == \tau$  Then

$k = 0$

Go To Break

Else

$k = 1$

End

End

Break: Continue

If  $k == 1$  Then

$$DM \leftarrow DM \cup \{\mathcal{M}\}$$

$$DMt \leftarrow DMt \cup \{\tau\}$$

End

Else

$$DM \leftarrow DM \cup \{\mathcal{M}\}$$

$$DMt \leftarrow DMt \cup \{\tau\}$$

End

End

End

End

### E) Algoritmelerde Kullanılan Notasyon ve Bunlarla İlgili Açıklamalar

$H$  sıralı vektörlerden oluşan bir kümeyi ve  $J$  de sıralı elemanlardan oluşan bir vektörü göstermek üzere;  $\{H\}_i$  :  $H$  kümesinin içinde bulunan  $i$ . ( $i \in \mathcal{N}$ ) vektörü,  $J(i)$  :  $J$  vektörünün içinde bulunan  $i$ . elemanı,  $mx(J)$  :  $J$  vektörünün içinde bulunan maksimum elemanı,  $max(H)$  :  $H$  kümesinin içindeki vektörlerin her birinin maksimum elemanının tutulduğu vektörün maksimum elemanının bulunması işlemi,  $min(H)$  :  $H$  kümesinin içindeki vektörlerin her birinin minimum elemanının tutulduğu vektörün minimum elemanının bulunması işlemi,  $length(\{H\}_i)$  :  $H$  kümesinin içinde bulunan  $i$ . vektörün uzunluğunun bulunması işlemi ( $length([\ ])$  = 0) göstermektedir. step  $Q$ , algoritmalar içerisindeki döngülerde zamanın  $Q$  ile gösterilen değer kadarlık adımlar halinde artırılmasını ifade etmektedir. Böylece,

$$\tau = \tau_0$$

Do Loop

$$\tau \leftarrow \tau + Q$$

If  $\tau == \tau_f$  Then

Exit Loop

End

End Loop

döngüsüne göre zaman  $\tau_f$  anına kadar  $Q$  ile belirtilen adımlar halinde artırılır.  $==$  gösterimi, algoritmalar içerisinde karşılaştırma işlemi yapılırken kullanılmaktadır ve eşitliği sorgulamaktadır. Karşılaştırılanlar eğer iki vektör ise eleman eleman karşılaştırma yapılmaktadır.  $\cup$ , kümeler için birleşim işlemi ifade etmektedir. Bu yapılırken elde edilen eleman mevcut kümenin elemanlarının sonuna ilave edilmektedir. Yani  $n$  tane elemana sahip bir kümeye bu şekilde yeni bir eleman ilave edilirse, bu yeni eleman bu kümenin  $n+1$ . elemanı olarak alınır.  $\hat{\cup}$ , vektörler için birleşim işlemi ifade etmektedir. Bu yapılırken yeni değer mevcut vektörün elemanlarının sonuna ilave edilmektedir. Örnek olarak  $A \hat{\cup} b$  işlemi sonucunda  $[A \mid b]$  vektörü elde edilmektedir.  $[\ ]$ , boş vektörü göstermektedir.  $\phi$ , içinde bulunan tüm vektörlerin boş olduğu kümeyi göstermektedir. Örneğin  $ET = \phi$  ise  $ET$  kümesi  $\{[\ ] [\ ] [\ ] \dots [\ ]\}$  şeklinde verilmektedir.

## F) Algortimaların Gerçeklemesiyle Elde Edilen MATLAB Programları

Bu kısımda oluşturulan MATLAB programları sırasıyla verilecektir. İlk olarak diğer fonksiyonları çağıran ana fonksiyonun içeriğine yer verilecektir. Hesaplama kolaylığı açısından UK-1 algoritmasında verilen işlemler iki farklı fonksiyon kullanılarak gerçekleştirilmiştir. Bunlar ana fonksiyondan sonra verilen "begin" ve "newmark" fonksiyonlarıdır.

Ardından verilen “cont” fonksiyonunda da UK-2 algoritmasında belirtilen işlemler gerçekleştirilmektedir. Daha sonra Sistem Çıkmazı algoritmasındaki işlemlerin gerçekleştirildiği “deadst” fonksiyonu verilmektedir. Son olarak ise “newmark” ve “cont” fonksiyonlarındaki bazı işlemlerin gerçekleştirildiği “test1” ve “test2” yardımcı fonksiyonları verilmektedir.

### ana fonksiyon

```
FILE=input('Lütfen girdi ve çıktı matrislerini, başlangıç işaretlemesini, her bir geçişe atanmış olan ateşlenme gecikmelerini, her bir geçiş için bekleme sürelerini ve başlangıç zamanını içeren dosyanın adını giriniz:','s');
```

```
load(FILE)
```

```
finaltime = input('Lütfen analizin sona erme anını giriniz:');
```

```
kmax = input('Lütfen izin verilen maksimum belirti sayısını giriniz:');
```

```
tic
```

```
if m0 <= kmax
```

```
    [p, t] = size(N);
```

```
    G = [], H = [], CE = [], Ft = [], RS = [], RSt = [],
```

```
    X = [], RSd = [], RStd = [], Ftd = [], COMP1 = [];
```

```
    A = cell(1, t), R = cell(1, t), Rdummy = cell(50000, t), Rd = cell(50000, t)
```

```
    DS = [], DSt = [], zaf = 1;
```

```
    [RS, RSt, Ft, Ma, o, f, A, kl, dummy, dummyzaman, Mdummy, Adummy,
```

```
    Rdummy, RSd, RStd, Ftd] = begin(currenttime, N, m0, xi, t, p, G, H, A, R, Rdummy, RSd, RStd, Ftd);
```

```
    if (Ft)' == zeros
```

```
        sprintf('Başlangıç işaretlemesinde ateşlenebilir geçiş bulunmamaktadır! Lütfen başlangıç işaretlemesini değiştirip analize tekrar başlayınız.')
```

```
    else
```

```
[RS, RSt, Ft, R, mt, currenttime, A, Mtemp, RSd, RStd, Ftd] =
newmark(Ma, N, currenttime, o, f, t, p, A, D, RS, RSt, Ft, CE, R,
RSd, RStd, Ftd);
```

```
[RS, RSt, Ft, RSd, RStd, Ftd, Rd, NN] =cont(mt, R, A, N, O, D,
kl, Mtemp, currenttime, finaltime, t, p, RS, RSt, Ft, dummy,
dummyzaman, Mdummy, Adummy, Rdummy, xi, zaf, COMP1, o, f,
Ma, RSd, RStd, Ftd, Rd, kmax);
```

```
RS
```

```
RSt
```

```
Ft
```

```
RSd
```

```
RStd
```

```
Ftd
```

```
if length(NN) >= 1
```

```
    NNN = max(NN);
```

```
    if max(RStd) <= NNN & NNN ~ = finaltime
```

```
        sprintf('Analizin sona erme anı maksimum %d olmalıdır!',
NNN)
```

```
    end
```

```
end
```

```
[DS, DSt] =deadst(DS, DSt, Ftd, Rd, RStd, RSd);
```

```
if length(DS) < 1
```

```
    disp('Girilen zaman aralığında Petri ağında sistem çıkmazı
oluşmamaktadır.')
```

```
else
```

```
    disp('Sistem çıkmazının oluştuğu işaretlemeler ve elde edilme
anları aşağıdaki gibidir.')
```

*DS*

*DSt*

**end**

**end**

**else**

sprintf('Lütfen başlangıç işaretlemenizi Petri ağındaki her bir yerde en fazla %d belirti olacak şekilde değiştirip analize tekrar başlayınız!',  $k_{max}$ )

**end**

toc

**begin fonksiyonu**

function[*aa, bb, cc, dd, ee, ff, gg, hh, ii, jj, kk, ll, mm, nn, oo, pp*]=begin(*currenttime, N, m<sub>0</sub>, xi, t, p, G, H, A, R, R<sub>dummy</sub>, RSd, RStd, Fid*)

*dummy* = [ ], *dummyzaman* = [ ]

*M<sub>dummy</sub>* = cell(1, 50000), *A<sub>dummy</sub>* = cell(50000, *t*);

*M<sub>a</sub>* = *m<sub>0</sub>*;

**for** *j* = 1 : *t*

*W* = [ ], *C* = [ ], *m* = [ ], *x* = [ ];

**for** *i* = 1 : *p*

*s* = 0;

**if** *M<sub>a</sub>*(1, *i*) < *N*(*i, j*)

*w* = 0;

*W* = [*W w*];

**else**

*w* = 1;

*W* = [*W w*];



```

if  $N(i, j) == 1$ 
   $s = 1$ ;
  for  $u = 1 : t$ 
    if  $N(i, u) == 1 \ \& \ u \sim j$ 
       $s = s + 1$ ;
    end
  end
end
if  $s == 1$ 
   $c = M_a(1, i)$ ;
   $C = [C \ c]$ ;
else
  if  $(M_a(1, i)/s) \leq 1$ 
     $c = 1$ ;
     $C = [C \ c]$ ;
  else
     $c = \text{ceil}(M_a(1, i)/s)$ ;
     $C = [C \ c]$ ;
  end
end
end
end
end
if  $W == \text{ones}$ 
   $m = \min(C)$ ;
   $X(j) = j$ ;
  for  $n = 1 : m$ 

```

```

     $x(n) = \text{round}((\text{currenttime} + ((n - 1) * xi(j))) * 100) / 100;$ 
     $A\{1, j\} = x;$ 
     $g = \text{length}(A\{1, j\});$ 
     $G = [G\ g];$ 
     $h = \text{max}(A\{1, j\});$ 
     $H = [H\ h];$ 

    end

     $f = \text{max}(G);$ 
     $o = \text{max}(H);$ 

else

     $X(j) = 0;$ 

end

end

 $U = [ ], kl = 0, zf = 0;$ 

for  $j = 1 : t$ 

    if  $X(j) \sim 0$ 

         $zf = zf + 1;$ 
         $U(zf) = X(j);$ 

    end

end

end

 $DE = [ ];$ 

for  $i = 1 : p$ 

     $tr = 0;$ 

    for  $j = 1 : t$ 

        if  $N(i, j) == 1$ 

             $tr = tr + 1;$ 

```

```

        end

    end

    if  $tr > 1$ 
         $de = i$ ;
         $DE = [DE\ de]$ ;
    end

end

 $EF = [ ]$ ;

for  $j = 1 : t$ 
    for  $i = 1 : p$ 
        if  $\text{ismember}(i, DE) == 1$ 
            if  $N(i, j) == 1$ 
                 $ef = j$ ;
                if  $\text{ismember}(ef, EF) == 0$ 
                     $EF = [EF\ ef]$ ;
                end
            end
        end
    end
end

end

end

if  $\text{length}(U) > 1$ 
    for  $j = 1 : t$ 
        if  $\text{min}(U) == j$ 
            for  $nt = j + 1 : t$ 
                if  $X(nt) \sim 0$ 
                    for  $i = 1 : p$ 

```



$ff = \max(G);$

$gg = A;$

$hh = kl;$

$ii = dummy;$

$jj = dummyzaman;$

$kk = M_{dummy};$

$ll = A_{dummy};$

$mm = R_{dummy};$

$nn = (m_0)';$

$oo = currenttime;$

$pp = X';$

### newmark fonksiyonu

function  $[aa, bb, cc, dd, ee, ff, gg, hh, ii, jj, kk] = \text{newmark}(M_a, N, o, f, t$   
 $currenttime, p, A, D, RS, RSt, Ft, CE, R, RSd, RStd, Ftd)$

$M_{temp} = M_a;$

$d = 1;$

**while**  $currenttime \leq o$

$K = [ ];$

**for**  $j = 1 : t$

$k = 0;$

**if**  $\text{length}(A\{1, j\}) \geq d$

**if**  $currenttime == A\{1, j\}(d)$

$W = [ ];$

**for**  $i = 1 : p$

**if**  $M_{temp}(1, i) < N(i, j)$

$w = 0;$

```

        W = [W w];
    else
        w = 1;
        W = [W w];
    end
end
end
if W == ones
    for i = 1 : p
        if N(i, j) == 1 & Mtemp(1, i) >= N(i, j)
            k = 1;
            Mtemp(1, i) = Mtemp(1, i) - 1;
        end
    end
end
if k == 1
    R {1, j} (d) = round((A {1, j} (d) + D(j)) * 100) / 100;
end
end
end
end
K = [K k];
end
if any(K) == 1
    Mnew = Mtemp;
    RS = [RS (Mnew)'];
    RSt = [RSt currenttime];
    [X] = test1(Mnew, N, p, t)
end

```

```

Rl = [Rl X'];
U = [ ], zf = 0;
for j = 1 : t
    if X(j) ≈ 0
        zf = zf + 1;
        U(zf) = X(j);
    end
end
DE = [ ];
for i = 1 : p
    tr = 0;
    for j = 1 : t
        if N(i, j) == 1
            tr = tr + 1;
        end
    end
    if tr > 1
        de = i;
        DE = [DE de];
    end
end
EF = [ ];
for j = 1 : t
    for i = 1 : p
        if ismember(i, DE) == 1
            if N(i, j) == 1

```

```

    ef = j;
    if ismember(ef, EF) == 0
        EF = [EF ef];
    end
end
end
end
end
if length(U) > 1
    for j = 1 : t
        if min(U) == j
            for nt = j + 1 : t
                if X(nt) ~ 0
                    for i = 1 : p
                        if N(i, nt) == 1
                            if (Mnew(1, i)/sum(N(i, :))) &
                                ismember(i, DE) == 1 &
                                ismember(nt, EF) == 1
                                kl = kl + 1;
                                dummy(1, kl) = nt;
                                dummyzaman(1, kl) = A {1, nt} (d+1);
                                Mdummy {1, kl} = Mnew;
                                for w = 1 : t
                                    Adummy {kl, w} = A {1, w};
                                    Rdummy {kl, w} = R {1, w};
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```





```

end
if currenttime == 0
    currenttime = round(currenttime * 100)/100;
    break
else
    currenttime = round((currenttime + 0.01) * 100)/100;
end
end
for j = 1 : t
    A{1, j} = [ ];
    ce = min(R{1, j});
    CE = [CE ce];
end
mt = min(CE);
aa = RS;
bb = RSt;
cc = Ft;
dd = R;
ee = mt;
ff = currenttime;
gg = A;
hh = Mtemp;
ii = RSd;
jj = RStd;
kk = Ftd;

```



```

                zz = R{1, j}(1);
            end
        end
    end
end

end

if k == 1
    COMP1 = [];
    Mnew = Mtemp;
    [ddd, eee] = size(RS);
    for uu = 1 : eee
        for ww = 1 : ddd
            comp(ww) = RS(ww, uu);
        end
        if comp == Mnew & RSt(uu) == currenttime
            comp1 = 1;
        else
            comp1 = 0;
        end
        COMP1 = [COMP1 comp1];
    end
end

if COMP1 == zeros
    if Mnew <= kmax
        RS = [RS (Mnew)'];
        RSd = [RSd (Mnew)'];
        RSt = [RSt currenttime];
    end
end

```

```

RStd = [RStd currenttime];

[X, kl, dummy, dummyzaman, Mdummy,
Adummy, Rdummy, Ma, o, f, A, R, RSt, Ft,
currenttime, COMP1] = test2(Mnew, N, p, t,
currenttime, kl, A, R, RSt, Ft, zz, xi, dummy,
dummyzaman, Mdummy, Adummy, Rdummy, f, o,
COMP1);

Ft = [Ft X'];

Ftd = [Ftd X'];

[kk, ll] = size(Ftd);

AB = [ ];

for j = 1 : kk
    Rd {ll, j} = R {1, j};
end

AB = (Ftd(:, ll))';

if AB == zeros

    GK = [ ];

    for j = 1 : t

        if length(R {1, j}) >= 1

            if max(R {1, j}) > currenttime

                gk = 1;

                GK = [GK gk];

            else

                gk = 0;

                GK = [GK gk];

            end

        end

    end

end

```

```

end
if  $GK == \text{zeros}$ 
     $nn = \text{currenttime}$ ;
     $NN = [NN \ nn]$ ;
     $\text{currenttime} = \text{finaltime}$ ;
     $\text{cikis} = 1$ ;
    break
end
end
end
else
     $[X, kl, \text{dummy}, \text{dummyzaman}, M_{\text{dummy}}, A_{\text{dummy}},$ 
     $R_{\text{dummy}}, M_a, o, f, A, R, RSt, F_t, \text{currenttime},$ 
     $COMP1] = \text{test2}(M_{\text{new}}, N, p, t, \text{currenttime}, kl,$ 
     $A, R, RSt, F_t, zz, xi, \text{dummy}, \text{dummyzaman},$ 
     $M_{\text{dummy}}, A_{\text{dummy}}, R_{\text{dummy}}, f, o, COMP1)$ ;
end
end
end
end
 $d = 1$ ;
while  $\text{currenttime} \leq o$ 
     $K = [ ]$ ;
    for  $j = 1 : t$ 
         $k = 0$ ;
        if  $\text{length}(A \{1, j\}) \geq d$ 
            if  $\text{currenttime} == A \{1, j\} (d)$ 

```

```

W = [ ];
for i = 1 : p
    if Mtemp(1, i) < N(i, j)
        w = 0;
        W = [W w];
    else
        w = 1;
        W = [W w];
    end
end
end
if W == ones
    for i = 1 : p
        if N(i, j) == 1 & Mtemp(1, i) >= N(i, j)
            k = 1;
            Mtemp(1, i) = Mtemp(1, i) - 1;
        end
    end
end
if k == 1
    RH(d) = round((A {1, j} (d) + D(j)) * 100) / 100;
    R {1, j} = [R {1, j} RH(d)];
end
end
end
end
K = [K k];
end

```

```

if any(K) == 1
    COMP1 = [ ];
    Mnew = Mtemp;
    [ddd, eee] = size(RS);
    for uu = 1 : eee
        for ww = 1 : ddd
            comp(ww) = RS(ww, uu);
        end
        if comp == Mnew & RSt(uu) == currenttime
            comp1 = 1;
        else
            comp1 = 0;
        end
        COMP1 = [COMP1 comp1];
    end
    if COMP1 == zeros
        RS = [RS (Mnew)'];
        RSt = [RSt currenttime];
        [X] = test1(Mnew, N, p, t);
        Ft = [Ft X'];
    U = [ ], zf = 0;
    for j = 1 : t
        if X(j) ~ = 0
            zf = zf + 1;
            U(zf) = X(j);
        end
    end

```



```
end
DE = [ ];
for i = 1 : p
    tr = 0;
    for j = 1 : t
        if N(i, j) == 1
            tr = tr + 1;
        end
    end
    if tr > 1
        de = i;
        DE = [DE de];
    end
end
EF = [ ];
for j = 1 : t
    for i = 1 : p
        if ismember(i, DE) == 1
            if N(i, j) == 1
                ef = j;
                if ismember(ef, EF) == 0
                    EF = [EF ef];
                end
            end
        end
    end
end
end
end
```

```

end

if length(U) > 1
    for j = 1 : t
        if min(U) == j
            for nt = j + 1 : t
                if X(nt) ~ = 0
                    for i = 1 : p
                        if N(i, nt) == 1
                            if (Mnew(1, i)/sum(N(i, :))) &
                                ismember(i, DE) == 1 &
                                ismember(nt, EF) == 1
                                    kl = kl + 1;
                                    dummy(1, kl) = nt;
                                    dummyzaman(1, kl) = A {1, nt} (d+1);
                                    Mdummy {1, kl} = Mnew;
                                    for w = 1 : t
                                        Adummy {kl, w} = A {1, w};
                                        Rdummy {kl, w} = R {1, w};
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
    for j = 1 : t
        if length (Adummy {kl, j}) > 1
            if currenttime == Adummy {kl, j} (1)
                B = [ ];
                for u = 1 :length(Adummy {kl, j})-1
                    B(u) = Adummy {kl, j} (u + 1);
                end
            end
        end
    end
end

```



```

end

if currenttime == 0
    currenttime = round(currenttime * 100)/100;
    break
else
    currenttime = round((currenttime + 0.01) * 100)/100;
end

end

end

CE = [ ];

for j = 1 : t
    if length(R {1, j}) >= 1
        if currenttime == R {1, j} (1)
            if length(R {1, j}) == 1
                R {1, j} = [ ];
            else
                B = [ ];
                for u = 1 : length(R {1, j}) - 1
                    B(u) = R {1, j} (u + 1);
                end
                R {1, j} = B;
            end
        end
    end

end

end

ce = min(R {1, j});

CE = [CE ce];

end

```

```

     $mt = \min(CE);$ 
    if length( $mt$ ) < 1
        break
    end

    if  $currenttime \geq finaltime$ 
        for  $j = 1 : t$ 
             $R\{1, j\} = [ ];$ 
        end

         $cikis = 1;$ 
    end

end

 $zaf$ 

 $kl$ 

if  $zaf \leq kl$ 
     $M_a = M_{dummy}\{1, zaf\};$ 
     $currenttime = dummyzaman(1, zaf);$ 
    for  $j = 1 : t$ 
         $A\{1, j\} = A_{dummy}\{zaf, j\};$ 
         $R\{1, j\} = R_{dummy}\{zaf, j\};$ 
    end

     $G = [ ];$ 
     $H = [ ];$ 
    for  $j = 1 : t$ 
         $g = \text{length}(A\{1, j\});$ 
         $G = [G\ g];$ 
         $h = \max(A\{1, j\});$ 
    end
end

```

```

        H = [H h];
    end

    f = max(G);
    o = max(H);

else

    exit = 1;

    break

end

Mtemp = Ma;
d = 1;

while currenttime <= o

    K = [ ];

    for j = dummy(1, zaf) : t;

        k = 0;

        if length(A {1, j}) >= d

            if currenttime == A {1, j} (d)

                W = [ ];

                for i = 1 : p

                    if Mtemp(1, i) < N(i, j)

                        w = 0;

                        W = [W w];

                    else

                        w = 1;

                        W = [W w];

                    end

                end

            end

        end

    end

end

```

```

if  $W == \text{ones}$ 
    for  $i = 1 : p$ 
        if  $N(i, j) == 1 \ \& \ M_{temp}(1, i) \geq N(i, j)$ 
             $k = 1$ ;
             $M_{temp}(1, i) = M_{temp}(1, i) - 1$ ;
        end
    end
end
if  $k == 1$ 
     $RH(d) = \text{round}((A\{1, j\}(d) + D(j)) * 100) / 100$ ;
     $R\{1, j\} = [R\{1, j\} \ RH(d)]$ ;
end
end
end
end
 $K = [K \ k]$ ;
end
for  $j = 1 : t$ 
     $k = 0$ ;
    if  $\text{length}(A\{1, j\}) \geq d$ 
        if  $\text{currenttime} == A\{1, j\}(d)$ 
             $W = [ ]$ ;
            for  $i = 1 : p$ 
                if  $M_{temp}(1, i) < N(i, j)$ 
                     $w = 0$ ;
                     $W = [W \ w]$ ;
                else

```

```

        w = 1;
        W = [W w];
    end
end
if W == ones
    for i = 1 : p
        if N(i, j) == 1 & M_temp(1, i) >= N(i, j)
            k = 1;
            M_temp(1, i) = M_temp(1, i) - 1;
        end
    end
    end
    if k == 1
        R {1, j} (d) = round((A {1, j} (d)+D(j))*100)/100;
    end
end
end
end
end
K = [K k];
end
if any(K) == 1
    COMP1 = [ ];
    M_new = M_temp;
    [ddd, eee] = size(RS);
    for uu = 1 : eee
        for ww = 1 : ddd
            comp(ww) = RS(ww, uu);

```



```

end
if comp ==  $M_{new}$  &  $RSt(uu) == currenttime$ 
    comp1 = 1;
else
    comp1 = 0;
end
COMP1 = [COMP1 comp1];
end
if COMP1 == zeros
    RS = [RS ( $M_{new}$ )'];
    RSt = [RSt currenttime];
    [X] = test1( $M_{new}$ , N, p, t);
    Ft = [Ft X'];
    U = [ ], zf = 0;
    for j = 1 : t
        if X(j) ~ 0
            zf = zf + 1;
            U(zf) = X(j);
        end
    end
end
DE = [ ];
for i = 1 : p
    tr = 0;
    for j = 1 : t
        if N(i, j) == 1
            tr = tr + 1;

```

```

        end

    end

    if  $tr > 1$ 
         $de = i$ ;
         $DE = [DE\ de]$ ;
    end

end

 $EF = [ ]$ ;

for  $j = 1 : t$ 
    for  $i = 1 : p$ 
        if  $\text{ismember}(i, DE) == 1$ 
            if  $N(i, j) == 1$ 
                 $ef = j$ ;
                if  $\text{ismember}(ef, EF) == 0$ 
                     $EF = [EF\ ef]$ ;
                end
            end
        end
    end
end

end

if  $\text{length}(U) > 1$ 
    for  $j = 1 : t$ 
        if  $\text{min}(U) == j$ 
            for  $nt = j + 1 : t$ 
                if  $X(nt) \sim 0$ 
                    for  $i = 1 : p$ 

```

```

if  $N(i, nt) == 1$ 
    if  $(M_{new}(1, i)/\text{sum}(N(i, :))) \&$ 
         $\text{ismember}(i, DE) == 1 \&$ 
         $\text{ismember}(nt, EF) == 1$ 
             $kl = kl + 1;$ 
             $\text{dummy}(1, kl) = nt;$ 
             $\text{dummyzaman}(1, kl) = A\{1, nt\}(d+1);$ 
             $M_{dummy}\{1, kl\} = M_{new};$ 
            for  $w = 1 : t$ 
                 $A_{dummy}\{kl, w\} = A\{1, w\};$ 
                 $R_{dummy}\{kl, w\} = R\{1, w\};$ 
            end
            for  $j = 1 : t$ 
                if  $\text{length}(A_{dummy}\{kl, j\}) > 1$ 
                    if  $\text{currenttime} == A_{dummy}\{kl, j\}(1)$ 
                         $B = [ ];$ 
                        for  $u = 1 : \text{length}(A_{dummy}\{kl, j\}) - 1$ 
                             $B(u) = A_{dummy}\{kl, j\}(u + 1);$ 
                        end
                         $A_{dummy}\{kl, j\} = B;$ 
                    end
                end
            end
            break
        end
    end

```

```

        end
    end
end
end
end
end
else
    if  $d == \text{length}(A\{1, j\})$ 
         $A\{1, j\} = []$ ;
    end
end
end
if  $d < f$ 
     $d = d + 1$ ;
end
if currenttime == 0
    currenttime = round(currenttime * 100)/100;
    break
else
    currenttime = round((currenttime + 0.01) * 100)/100;
end
end
end
CE = [ ];
for  $j = 1 : t$ 
    if length( $R\{1, j\}$ ) >= 1
        if currenttime ==  $R\{1, j\}(1)$ 

```

```

    if length(R {1, j}) == 1
        R {1, j} = [ ];
    else
        B = [ ];
        for u = 1 :length(R {1, j}) - 1
            B(u) = R {1, j} (u + 1);
        end
        R {1, j} = B;
    end

end

end

end

A {1, j} = [ ];
ce = min(R {1, j});
CE = [CE ce];

end

mt = min(CE);
zaf = zaf + 1;

end

aa = RS;
bb = RSt;
cc = Ft;
dd = RSd;
ee = RStd;
ff = Ftd;
gg = Rd;
hh = NN;

```

**deadst fonksiyonu**

```
function[aa, bb] = deadst(DS, DSt, Ftd, Rd, RStd, RSd)
```

```
[ggg, sss] = size(Ftd);
```

```
AB = cell(1, sss);
```

```
for i = 1 : sss
```

```
    AB {1, i} = (Ftd(:, i))';
```

```
end
```

```
for i = 1 : sss
```

```
    if AB {1, i} == zeros
```

```
        GK = [ ];
```

```
        for j = 1 : ggg
```

```
            if length(Rd {i, j}) >= 1
```

```
                if max(Rd {i, j}) > RStd(i)
```

```
                    gk = 1;
```

```
                    GK = [GK gk];
```

```
                else
```

```
                    gk = 0;
```

```
                    GK = [GK gk];
```

```
            end
```

```
        end
```

```
    end
```

```
    if GK == zeros
```

```
        ds = RSd(:, i);
```

```
        dst = RStd(i);
```

```
        if length(DS) < 1
```

```
            DS = ds;
```

```
DSt = dst;  
  
else  
  
    COMP1 = [ ];  
  
    [gg, zz] = size(DS);  
  
    for uu = 1 : zz  
  
        comp = DS(:, uu);  
  
        if comp == ds & DSt(uu) == dst  
  
            comp1 = 1;  
  
        else  
  
            comp1 = 0;  
  
        end  
  
        COMP1 = [COMP1 comp1];  
  
    end  
  
    if COMP1 == zeros  
  
        DS = [DS ds];  
  
        DSt = [DSt dst];  
  
    end  
  
end  
  
end  
  
end  
  
end  
  
aa = DS;  
  
bb = DSt;
```

**test1 fonksiyonu**

```
function[aa] = test1( $M_{new}, N, p, t$ )
```

```
 $X = [ ]$ ;
```

```
for  $j = 1 : t$ 
```

```
     $W = [ ]$ ;
```

```
    for  $i = 1 : p$ 
```

```
        if  $M_{new}(1, i) < N(i, j)$ 
```

```
             $w = 0$ ;
```

```
             $W = [W \ w]$ ;
```

```
        else
```

```
             $w = 1$ ;
```

```
             $W = [W \ w]$ ;
```

```
        end
```

```
    end
```

```
    if  $W == \text{ones}$ ;
```

```
         $X(j) = j$ ;
```

```
    else
```

```
         $X(j) = 0$ ;
```

```
    end
```

```
end
```

```
 $aa = X$ ;
```

**test2 fonksiyonu**

```
function[aa, bb, cc, dd, ee, ff, gg, hh, ii, jj, kk, ll, mm, nn, oo, pp] = test2( $N,$ 
```

```
 $M_{new}, p, t, currenttime, kl, A, R, RSt, Ft, zz, xi, dummy, dummyzaman,$ 
```

```
 $M_{dummy}, A_{dummy}, R_{dummy}, COMP1, f, o$ )
```

```
 $G = [ ], H = [ ], X = [ ]$ ;
```



```

 $M_a = M_{new};$ 
for  $j = 1 : t$ 
     $W = [ ], C = [ ], m = [ ], x = [ ];$ 
    for  $i = 1 : p$ 
         $s = 0;$ 
        if  $M_a(1, i) < N(i, j)$ 
             $w = 0;$ 
             $W = [W \ w];$ 
        else
             $w = 1;$ 
             $W = [W \ w];$ 
            if  $N(i, j) == 1$ 
                 $s = 1;$ 
                for  $u = 1 : t$ 
                    if  $N(i, u) == 1 \ \& \ u \sim j$ 
                         $s = s + 1;$ 
                    end
                end
            end
            if  $s == 1$ 
                 $c = M_a(1, i);$ 
                 $C = [C \ c];$ 
            else
                if  $(M_a(1, i)/s) \leq 1$ 
                     $c = 1;$ 
                     $C = [C \ c]$ 
                else

```

```

        c = ceil( $M_a(1, i)/s$ );
        C = [C c]
    end
end
end
end
end
if W == ones;
    m = min(C);
    X(j) = j;
    for n = 1 : m
        x(n) = round((zz + ((n - 1) * xi(j))) * 100)/100;
        A {1, j} = x;
        g = length(A {1, j});
        G = [G g];
        h = max(A {1, j});
        H = [H h];
    end
    f = max(G);
    o = max(H);
else
    X(j) = 0;
end
end
COMP1 = [ ];
[ddd, eee] = size(Ft);

```

```

for uu = 1 : eee - 1
    for ww = 1 : ddd
        comp(ww) = Ft(ww, uu);
    end
    if comp == X & RSt(uu) == currenttime
        comp1 = 1;
    else
        comp1 = 0;
    end
    COMP1 = [COMP1 comp1];
end
DE = [ ];
for i = 1 : p
    tr = 0;
    for j = 1 : t
        if N(i, j) == 1
            tr = tr + 1;
        end
    end
    end
    if tr > 1
        de = i;
        DE = [DE de];
    end
end
end
EF = [ ];
for j = 1 : t

```





$$ff = A_{dummy}$$

$$gg = R_{dummy}$$

$$hh = M_a;$$

$$ii = o;$$

$$jj = f;$$

$$kk = A;$$

$$ll = R;$$

$$mm = currenttime;$$

$$nn = COMP1;$$

$$oo = RSt;$$

$$pp = F_t;$$

### G) Farklı ZPA'larda Gerçekleştirilen Analizlerin Sonuçları

Şekil 5.1.'deki ZPA'da  $[0, 10]$  b.z. aralığında gerçekleştirilen analizde elde edilen  $RS$ ,  $RSt$  ve  $F_t$  kümeleri şu şekilde verilmektedir:

$$RS = \left\{ \begin{array}{l} \left[ \begin{array}{c} 1 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 1 \\ 3 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{array} \right] \end{array} \right\},$$











$$\left. \begin{array}{c} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right\}$$

$$RSt = \{0, 0, 1, 2, 2, 5, 5, 7, 7, 10, 10, 12, 12, 15, 15, 17, 17, 19, 19, 20, 20, 10, 12, \\ 12, 13, 13, 15, 15, 18, 18, 20, 20, 11, 12, 12, 15, 15, 17, 17, 19, 19, 18, 20, \\ 20, 14, 18\}$$

$$F_t = \left\{ \begin{array}{c} \left[ \begin{array}{c} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right], \left[ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \end{array} \right\},$$





$$DM = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$DMt = \{14, 18\}$$

Şekil 5.2.'deki ZPA'da geçişlere atanmış olan ateşlenme gecikmelerinin kümesinin  $D = [1 \ 2 \ 2 \ 1 \ 2 \ 2 \ 1 \ 2 \ 3 \ 1 \ 2]$  bz. olarak değiştirilmesinden sonra  $[0, 20]$  bz. aralığında gerçekleştirilen analizde elde edilen  $DM$  ve  $DMt$  kümeleri şu şekilde verilmektedir:

$$DM = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$DMt = \{11, 14, 20, 17\}$$

*D* kümesinde gerçekleştirilen deęişiklik sonucunda ateşlenmelerin daha kısa sürede tamamlanması sağlanmış ve bu sayede  $[0, 20]$  bz. aralığında daha fazla işaretleme vektörü elde edilmiştir. *DM* kümesinde de daha önce 2 adet işaretleme vektörü bulunuyor iken, deęişiklikten sonra ise sistem çıkmazının oluştuęu 4 adet işaretleme vektörü bulunmaktadır.