

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

SÜREÇ ANALİZİ VE KEŞFİ İÇİN BİR SÜREÇ MADENCİLİĞİ SİSTEMİ

MECİT YÜZKAT

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
DOÇ. DR. FETHULLAH KARABİBER**

İSTANBUL, 2016

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SÜREÇ ANALİZİ VE KEŞFİ İÇİN BİR SÜREÇ MADENCİLİĞİ SİSTEMİ

Mecit YÜZKAT tarafından hazırlanan tez çalışması 26.04.2016 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Fethullah KARABİBER
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Fethullah KARABİBER
Yıldız Teknik Üniversitesi

Doç. Dr. Şadi Evren ŞEKER
İstanbul Şehir Üniversitesi

Yrd. Doç. Dr. Mehmet Sıddık AKTAŞ
Yıldız Teknik Üniversitesi

ÖNSÖZ

Bu tezin hazırlanma sürecinde üstün bilgi birikimi ve yönlendirmeleriyle ufkumu açan, anlayışlı tutumuyla bana her türlü desteği veren Doç. Dr. Fethullah KARABİBER'e en içten teşekkürlerimi sunarım.

Öğrencilerimizden Buğra ŞEN, Ümmet ÇİVİ, Hasan Kaan CAYMAZ, Hüseyin MERAL ve asistan arkadaşım Çiğdem BAKIR'a katkılarından dolayı teşekkür ederim.

Bu tezi, destekleri ve bana olan inançları sayesinde karşılaştığım zorlukları aşmamda çok büyük katkıları olan sevgili aileme, sevgili eşime ve sevgili oğlum Miran Batın YÜZKAT'a ithaf ediyorum.

Nisan, 2016

Mecit YÜZKAT

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....	vi
KISALTMA LİSTESİ.....	vii
ŞEKİL LİSTESİ.....	viii
ÇİZELGE LİSTESİ	x
ÖZET	xi
ABSTRACT.....	xiii
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	3
1.3 Orijinal Katkı.....	4
BÖLÜM 2	
SÜREÇ MADENCİLİĞİ.....	5
2.1 Süreç Madenciliği Aşamaları.....	6
2.1.1 Süreçlerin Keşfi	7
2.1.2 Süreçlerin Uygunluk Kontrolü.....	7
2.1.3 Süreçlerin Geliştirilmesi	7
2.2 Örnek Süreç Olay Kaydının İncelenmesi	8
2.3 Süreç Madenciliğinde Kullanılan Araçlar	10
BÖLÜM 3	
SÜREÇ MADENCİLİĞİ KEŞİF ALGORİTMALARI	12
3.1 Alfa Algoritması.....	12
3.1.1 Alfa Algoritması Adımları.....	15
3.1.2 Alfa Algoritması için Örnek Uygulama	16

3.2	Sezgisel Algoritma	18
3.2.1	Sezgisel Algoritma Adımları	19
3.2.2	Sezgisel Algoritma için Örnek Uygulama	20
3.3	Genetik Algoritma	24
3.3.1	Genetik Algoritmanın Genel Tanımları	25
3.3.2	Genetik Algoritmanın Genel Adımları	25
3.3.3	Genetik Algoritmanın İşlemleri.....	26
3.3.3.1	Kodlama	27
3.3.3.2	Seçme	27
3.3.3.3	Çaprazlama.....	29
3.3.3.4	Mutasyon	30
3.3.4	Genetik Algoritma ile Süreç Keşfi	31
3.3.4.1	Genetik Algoritma için Nedensel Matris (Causal Matrix)	33
3.3.4.2	Genetik Algoritmada Ayrıştırma (Parsing) İşlemi.....	35
3.3.4.3	Genetik Algoritmada Çaprazlama	38
3.4	Süreç Madenciliği Keşif Algoritmalarının Karşılaştırma Kriterleri.....	41
BÖLÜM 4		
SÜREÇ MADENCİLİĞİ SİSTEMİNİN GERÇEKLENMESİ.....		43
4.1	Süreç Modeli	43
4.2	Simülasyon	45
4.3	Analiz.....	46
4.4	Takım Karşılaştırması	46
BÖLÜM 5		
DENEYSEL SONUÇLAR		48
5.1	Veri Üretici Simülasyon Sonuçları.....	48
5.2	Süreç Madenciliği Algoritmalarıyla Süreç Keşfi	54
5.3	Süreç Madenciliği Keşif Algoritmalarının Performans Analizi	56
BÖLÜM 6		
SONUÇ.....		60
KAYNAKLAR		62
ÖZGEÇMİŞ		65

SİMGE LİSTESİ

$>$	Doğrusal izleme
$\rightarrow$	Nedensellik
$\parallel$	Paralellik
$\#$	Seçme
P_W	İş akış ağı yerleri
T_W	İş akış ağı geçişleri
F_W	İş akış ağı yer ve geçişlerinin birleştirilmesi
C_{in}	Giriş değeri
C_{out}	Çıkış değeri

KISALTMA LİSTESİ

BPMN	Business Process Model Notation
CRM	Customer Relationship Management
EPC	Event-driven Process Chain
ERP	Enterprise Resource Planning
MXML	Mining Extensible Markup Language
ProM	Process Mining
UML	Unified Modeling Language
WFMS	Work Flow Management System
XES	Extensible Event Stream
XML	Extensible Markup Language

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1 Süreç Madenciliğinin iş zekası ilişkisi[15]	6
Şekil 2. 2 Süreç madenciliğinin üç adımı [15]	6
Şekil 2. 3 Süreç madenciliğinin keşif aşaması.....	7
Şekil 2. 4 Süreç madenciliğinin uygunluk kontrolü aşaması.....	7
Şekil 2. 5 Süreç madenciliğinin geliştirme aşaması	8
Şekil 2. 6 Örnek sipariş uygulaması [13].....	8
Şekil 2. 7 Sipariş örneğın petri ağı ile gösterimi [13]	10
Şekil 3. 1 x ve y aktiviteleri arasında nedensel ilişki gösterimi($x \rightarrow y$)	12
Şekil 3. 2 y ve z aktiviteleri arasındaki paralellik gösterimi($y \parallel z$).....	13
Şekil 3. 3 x ve y aktiviteleri arasındaki paralellik gösterimi($x \parallel y$)	13
Şekil 3. 4 y ve z aktiviteleri arasındaki seçme işleminin gösterimi($y \# z$)	14
Şekil 3. 5 x ve y aktiviteleri arasındaki seçme işleminin gösterimi($x \# y$).....	14
Şekil 3. 6 Yer, geçiş ve halka gösterimi.....	15
Şekil 3. 7 L1 olay kaydının petri ağı.....	18
Şekil 3. 8 Frekans değeri 2'den büyük olan aktivite grupları	22
Şekil 3. 9 Frekans değeri 5'den büyük olan aktivite grupları	22
Şekil 3. 10 Sezgisel algoritmaya göre oluşan petri ağı	23
Şekil 3. 11 Sezgisel algoritma ile elde edilen yolların görselleştirilmesi	24
Şekil 3. 12 Genetik algoritmanın adımları [29]	26
Şekil 3. 13 Kodlama çeşitleri[28]	27
Şekil 3. 14 Rulet çemberi.....	28
Şekil 3. 15 Rastgele seçme işlemine göre bireylerin seçim oranları	29
Şekil 3. 16 Çaprazlama metotları [30]	30
Şekil 3. 17 Çizelge 3.5 göre oluşan petri ağı.....	33
Şekil 3. 18 Rastgele üretilen iki bireyin petri ağı.....	39
Şekil 3. 19 Çaprazlama işleminin gerçekleştirilmesi.....	40
Şekil 4. 1 Örnek Kredi kartı başvuru süreci	43
Şekil 4. 2 Süreç oluşturma ekranı	44
Şekil 4. 3 Simülasyon sonucu üretilen bilgiler	45
Şekil 4. 4 Analiz sonucu çıkarılan özet bilgiler(sol genel ve sağ takım)	46
Şekil 4. 5 Adım Frekans analiz verileri	47
Şekil 5. 1 Örnek Hasta Takip Süreç modeli oluşturma	48
Şekil 5. 2 Oluşturulan Hasta Takip Süreç modelinin genel özeti.....	49
Şekil 5. 3 Oluşturulan Hasta Takip Süreç modelinin simülasyonunu oluşturma.....	49

Şekil 5. 4	Oluşturulan Hasta Takip Süreç modelinin simülasyon özeti.....	50
Şekil 5. 5	Hasta Takip Süreç modeline ait analiz sonuçları.....	50
Şekil 5. 6	Hasta Takip Süreç modeline ait genel frekans bilgileri	50
Şekil 5. 7	Hasta Takip Süreç modelinin 3.takıma ait frekans bilgileri.....	51
Şekil 5. 8	Hasta Takip Süreç modeline ait genel zaman frekans grafiği	51
Şekil 5. 9	Hasta Takip Süreç modelinin 3.takımına ait zaman frekans grafiği.....	51
Şekil 5. 10	Hasta Takip Sürecinin 3.takımının 44.track'e ait zaman frekans grafiği.....	52
Şekil 5. 11	Hasta Takip Süreç modeline ait genel maliyet grafiği	52
Şekil 5. 12	Hasta Takip Süreç modelinin 3.takıma ait maliyet grafiği	53
Şekil 5. 13	Hasta Takip Sürec modelinin 3.takımının 44.track'e ait maliyet grafiği	53
Şekil 5. 14	Hasta Takip Süreç modeline ait takımlar arası karşılaştırma grafiği.....	54
Şekil 5. 15	Örnek kredi veri setinin alfa algoritmasıyla oluşturulmuş petri ağı	54
Şekil 5. 16	Örnek kredi veri setinin sezgisel algoritma ile oluşturulmuş petri ağı	55
Şekil 5. 17	Örnek kredi veri setinin genetik algoritma ile oluşturulmuş petri ağı.....	55
Şekil 5. 18	Hasta Takip Sürecinin görselleştirilmesi	56

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2. 1 Sipariş örneğinin olay kayıt tablosu	9
Çizelge 2. 2 Süreç madenciliğinde kullanılan araçlar [13]	11
Çizelge 3. 1 L1 olay kaydının izleri	17
Çizelge 3. 2 L2 olay kaydının izleri	20
Çizelge 3. 3 L2 olay kaydının nedensellik ilişkisinin hesaplanması	21
Çizelge 3. 4 Bireylerin mutasyon işlemi.....	31
Çizelge 3. 5 Örnek bir olay kaydı	32
Çizelge 3. 6 Nedensellik matrisi.....	34
Çizelge 3. 7 Nedensel matrisin kodlanmış kısa ve öz hali.....	35
Çizelge 3. 8 Durum 3 için ayrıştırma adımları.....	36
Çizelge 3. 9 Rastgele üretilen iki birey.....	39
Çizelge 3. 10 Çaprazlama sonucu oluşan iki yavru birey	40
Çizelge 5. 1 Süreç madenciliği algoritmalarının simülasyon veri setleri üzerindeki sonuçları	57
Çizelge 5. 2 Süreç madenciliği algoritmalarının mevcut sentetik veri setleri üzerindeki sonuçları	58
Çizelge 5. 3 Süreç madenciliği algoritmalarının gerçek veri setleri üzerindeki sonuçları	59

SÜREÇ ANALİZİ VE KEŞFİ İÇİN BİR SÜREÇ MADENCİLİĞİ SİSTEMİ

Mecit YÜZKAT

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Fethullah KARABİBER

Süreç madenciliği, iş dünyasının verimlilik ve üretkenliği arttırma ihtiyacını karşılamayı amaçlayan yeni bir çalışma alanıdır. Bu alan iş süreçlerini analiz, keşfetme, yönetme ve iyileştirme konusunda çalışmalar yapar. Süreç madenciliği, olay kayıtlarını kaynak olarak kullanır ve bu kaynağın üzerinde çalışır. Böylece süreç modelindeki her adımı içeren olay kayıtları analiz edilerek sistem geliştirilir.

Tez çalışması süreç analizi ve keşfi olmak üzere iki önemli aşamadan oluşur. İlk aşamasında süreç modelleme ve geliştirilmesine yardımcı olabilecek veri üretebilen bir simülatör geliştirmesi amaçlanmıştır. Bu çalışma kapsamında çeşitli süreç modelleri ile çalışabilen ve bu modellerden anlamlı bilgiler çıkaran bir sistem gerçekleştirilmiştir. Bu sistem sayesinde daha üretken ve verimli süreçler geliştirilebilir. Simülatör dört modülden oluşmaktadır. İlk modül, kullanıcıların iş süreç modeli oluşturduğu bölümdür. Bu modülde kullanıcı, sistemin arayüzünde kendi iş süreci modelini oluşturabilir ya da kayıtlı diğer modellerden seçim yapabilir. İkinci modülde ise oluşturulan bu süreç modelleri kullanılarak takım bazlı verilerin simülasyonu yapılır. Analiz olarak isimlendirilen üçüncü modülde üretilen bu veriler kullanılarak anlamlı bilgiler ortaya çıkartılır. Son modül olan takım karşılaştırmasında takımlar arasındaki performans dağılımları karşılaştırılır. Sonuç olarak, üretilen verilerdeki zaman, kaynak ve maliyet bilgilerinden yola çıkılarak oluşturulan sürecin iyileştirilmesi amaçlanmıştır.

Tezin ikinci aşamasında süreç madenciliği keşif algoritmaları ile sentetik ve gerçek olay kayıtları kullanılarak süreç keşfi yapılmıştır. Keşfedilen süreçler petri ağları ile gösterilerek algoritmaların başarımları uygunluk fonksiyonu, doğruluk oranları ve

alıřma sreleri kullanarak karřılařtırılmıřtır. Sonu olarak; sezgisel algoritma alfa algoritmasının dezavantajlı yanları olan verideki grlt ve tamamlanmayan sreleri geliřtirdiėinden dolayı daha bařarılıdır. Genetik algoritmada ise genetik operatrler(seme, aprazlama, mutasyon) kullanıldıėından dolayı alfa ve sezgisel algoritmaya gre daha bařarılı sonular vermiřtir.

Anahtar Kelimeler: Sre madenciliėi, veri reteci, petri aėları, alfa algoritması, sezgisel algoritma, genetik algoritma

ABSTRACT

A PROCESS MINING SYSTEM FOR PROCESS ANALYSIS AND DISCOVERY

Mecit YÜZKAT

Department of Computer Engineering

MSc. Thesis

Adviser: Assoc. Prof. Dr. Fethullah KARABİBER

The process mining is a new field which aims at satisfying the need for productivity and efficiency in the business world. It focuses on the analysis, discovery, management and improvement of the business processes. Process mining uses the event logs as a source and works on this source. As such the event logs consisting of each step in the process model is analysed and developed.

The thesis study includes two important stages; one being the process analysis and the discovery. In the first stage development of a simulator which generates data to help in process modelling and development is intended. During this stage a system operating with various process models and deriving significant data from such models has been set up. More productive and efficient processes may be developed with this system. The simulator consists of four modules. The first one is the section where users set the business process model. In this module, the user may develop his own business process model in the interface of the system or he may choose among the other models stored. In the second module the already set up process models are used and the simulation of the team based data is made. In the third module, which may be named as analysis, the generated data are used and significant data is produced. The last module, which is the comparison of teams, the performance distributions among the teams are compared. As a result, the improvement of the process which is set up based on time, source and financial data is intended.

In the second stage of the dissertation the process discovery took place through the use of the discovery algorithms of process mining and real event records. The discovered processes have been displayed with the petri nets and they have been compared through the use of the performance of algorithms, function of fitness, accuracy rates and operating time. As a result , the heuristic algorithm is more successful as it develops the noise and incomplete processes in the data, which is the disadvantage of the alpha algorithm. And as genetic operators are used in genetic algorithm (selection, crossover and mutation) it has produced better results when compared with alpha and heuristic algorithm.

Keywords: process mining, data generator, petri nets, alpha algorithm, heuristic algorithm, genetic algorithm

1.1 Literatür Özeti

Son yıllarda süreç madenciliği ile ilgili çalışmalarda artış görülmekte ve bu konunun önemi her geçen gün artmaktadır. Süreç madenciliği konusunda daha önce yapılan benzer çalışmalara ait özet bilgilerden aşağıda kısaca bahsedilmektedir.

Rabin ve Aalst tüm süreç madenciliği algoritmalarını tek bir platformda toplamak için ProM yazılımını geliştirmişlerdir [1]. Bu yazılımla süreçleri keşfetmek, süreçlerin uygunluğunun ölçülmesi ve süreçlerinin iyileştirilmesi sağlanabilmektedir. ProM Genetik, Fuzzy, Sezgisel ve Alfa gibi birçok süreç madenciliği algoritmalarının birbiriyle ilişkili olarak çalıştığını gösteren bir platform sunar.

İncelenen diğer bir çalışmada Mans ve Schonenberg [2], süreç madenciliği teknikleri ile felçli hasta bakımı uygulamaları geliştirmişlerdir. Bu çalışmada, farklı hastanelerdeki felç geçirmiş hastaların tedavi süreçlerini keşfetmek için süreç madenciliğine başvurmuşlardır. Hastaların ilişkili oldukları olayları inceleyerek analizler ortaya koymuşlardır. Bu çalışma süreç madenciliği tekniklerinin başarılı bir biçimde farklı hastane ve farklı hastalara ait klinik verilerde kullanılabileceğini göstermektedir.

Stijn ve Jochen telekom sektöründeki olay kayıtlarını inceleyerek yaptıkları çalışmalarında, bilgi sistemleri olay kayıtlarından süreç keşfi yapmışlardır [3]. Olay kayıtlarından yapılan analizler sayesinde otomatik olarak süreçlerin oluşturulabileceği ve anlamlı verilerin çıkartılabileceği sonucu çıkarılmıştır.

Weerapong, Porouhan ve Premchaiswadi tarafından Taylan üniversitesindeki öğrencilerin kayıtlarından alınan 569 olay (event) ve 299 duruma (case) sahip veri kaydı ProM kullanılarak alfa algoritmasıyla analiz edilmeye çalışılmıştır [4]. Bu çalışmanın amacı ardışık olaylardan bilgi çıkarımını yeniden düzenlemektir. Ancak verinin gürültülü olması ve tamamlanmayan durumlardan dolayı istenilen başarı tam olarak sağlanamamıştır.

Weijters ve Aalst süreç madenciliğinde sezgisel algoritma kullanarak 12.000 farklı olay kaydından süreç keşfi ve analizini gerçekleştirmişlerdir. Yapılan çalışmada gürültülü veri ve açıkça belirtilmeyen süreçler üzerinde çalışılmıştır [5].

Saravanan ve Rama Sree, dinamik ve karmaşık boyalama süreçlerinin keşfi ve iş akış yönetiminin sağlanması için sezgisel algoritma kullanmışlardır [6]. Amaçları zaman maliyetini azaltarak nitelikli ve etkili bir süreç analizinin gerçekleştirilmesidir. Bu çalışmada ProM platformu üzerindeki sezgisel algoritma ile DWS (Disjunctive Workflow Scheme) kullanmışlardır.

Koonce ve Tsari genetik algoritma kullanarak üretimde iş idaresi için bir program oluşturmaya çalışmışlardır [7]. Bu çalışmanın amacı verilen çeşitli işlere karşı gerçekleştirilmesi mümkün olan çeşitli operasyonlardan en kısa işlem zamanının bulunarak optimum iş sürecinin belirlenmesidir. Yapılan çalışmada çeşitli kurallar oluşturularak farklı işlem zamanları için farklı operasyonlar oluşturulmuştur. Ancak genetik algoritmanın karşılaştığı performans problemi bu çalışmada tam olarak ele alınamamıştır.

Süreç madenciliğinin sağlık alanındaki uygulamalarından biride Avustralya'da dört farklı hastaneden alınan olay kayıtlarıyla süreç madenciliği analizi yapılarak hastalarla ilgili önemli bilgilerin elde edilmesi sağlanmıştır. Yapılan çalışmaların sonuçları ise karşılaştırılmalı olarak verilmeye çalışılmıştır. Ancak süreç madenciliğinin zorluklarından biri de ikiden fazla çoklu verilerde analiz etmenin, veriyi bulmanın ve iyileştirmenin oldukça zor olmasıdır. Hastalarla ilgili tüm bilgilerin bir arada bulunması büyük verilerle ilgili karşılaştırmalı analiz etmek, veriyi görselleştirmek ve veriyi bölmek birtakım sıkıntıları da beraberinde getirmektedir. Bu çalışmada kısmen de olsa bu eksiklikler

giderilmeye çalışılmıştır. Ancak hastalarla ilgili kategorik bilgiler hesaba katılmadığı için tam olarak başarı sağlanmamıştır [8].

Liu ve Zhang tarafından yapılan çalışmada [9], operasyonel karar destek sistemleri için olay grafikleri kullanarak birbiriyle alakalı süreçlerin entegrasyonları yapılmıştır. Farklı senaryolara ait üretilen olay kayıtlarından analizler yapılmıştır. Ulaşılan sonuçlar bize olay kayıt analizlerinin süreç yönetimi ve geliştirmesi açısından önemli rol oynadığını göstermektedir.

YAWL tüm iş akış sistemlerini birleştirip, sonuçlarını görselleştiren bir iş akış sistemidir [10]. Bu sistemde coğrafik haritalar, süreç haritaları ve organizasyon haritaları ile ilgili analizler yapılmaktadır. Bu çeşitli haritalar üzerinde çeşitli kaynaklar ve iş özellikleri arasındaki uzaklıklara bağlı olarak görselleştirme olanağı sunulmaktadır.

1.2 Tezin Amacı

Çağımızda bilgi işlem sistemleri iş dünyasında çok önemli bir rol oynamaktadır. İşletmeler için hızlı kararlar almak ve verimliliği artırmak büyük bir öneme sahiptir. Bu aşamada işletmedeki süreçlerin incelenmesi ile çalışanların performansı ve süreçteki adımların maliyeti gibi sonuçlara ulaşılmaktadır. Süreçlerdeki adımlar detaylı olarak incelenerek, adımların tamamlanma süresi ve maliyeti gibi bilgilere ulaşarak süreçteki sorunlu kısımlar ortaya çıkarılmakta ve bu sürecin verimliliği artırma işlemleri yapılabilmektedir.

Tezin ilk adımında süreç modellemesi, simülasyonu ve analizini gerçekleştirecek bir veri üretici oluşturulması amaçlanmıştır. Tezin ikinci adımında ise veri üretici yardımıyla oluşturulan sentetik olay kayıtları ve günümüz bilgi sistemlerinde bulunan gerçek olay kayıtları kullanılarak süreçlerin keşfinin yapılması hedeflenmiştir. Süreçlerin keşfi için alfa, sezgisel ve genetik algoritmalar gibi birden fazla süreç madenciliği keşif algoritmaları kullanılmıştır. Algoritmaların başarı oranları uygunluk fonksiyonu, doğruluk oranları ve çalışma sürelerine göre karşılaştırılmıştır. Kısacası bu tez çalışmasında hem süreç verisi üretebilen ve sentetik veya gerçek süreç verileri için süreç analizi, keşfini yapan bir süreç madenciliği sistemi geliştirmesi amaçlanmıştır.

1.3 Orijinal Katkı

Süreç madenciliği: model oluşturma, model geliştirmek, simülasyon modellerinin otomatikleştirilmesi, süreç keşfi, analizi, olay kayıtlarının karşılaştırılması ve geçmişe ait süreçlerden tahminlerde bulunmak gibi konuları içermektedir [11].

Günümüzde iş süreçleri birçok alanda yaygın olarak kullanılmaktadır. Bu çalışmada olay kayıtlarından elde edilen petri ağlarına göre aktiviteler ve aralarındaki ilişkiler hakkında bilgi edinmek amacıyla ortaya yeni bir yaklaşım sunuldu. Ayrıca iş süreçlerinin uygunluğunu ölçmek için kullanılan geleneksel algortimalarının dışında süreç madenciliği keşif algoritmaları kullanılmıştır.

Süreç madenciliğinin ülkemizde yeni bir alan olması ayrıca bu alanda bilimsel çalışmaların olmaması tez çalışma aşamasında bu eksikliği gidermek amacıyla süreç modelleme, simülatörü, analizini ve keşfini yapan süreç madenciliği sistemi geliştirilmiştir. Bu sistemle bu alanda gelecekte ülkemizde yapılacak çalışmalara bir referans gösterilmesi hedeflenmektedir.

Bu tez çalışmasında yapılan katkılar aşağıda maddeler halinde kısaca verilmektedir.

- Örnek bir süreç modeli için farklı durum ve senaryolar ile sentetik veri üretilmesi
- Süreçlerin akışlarının görsel olarak incelenmesi
- Yaygın kullanılan süreç madenciliği algoritmalarının gerçekleştirilmesi
- Süreç madenciliği alanında kullanılabilecek bir yazılım platformunun geliştirilmesi

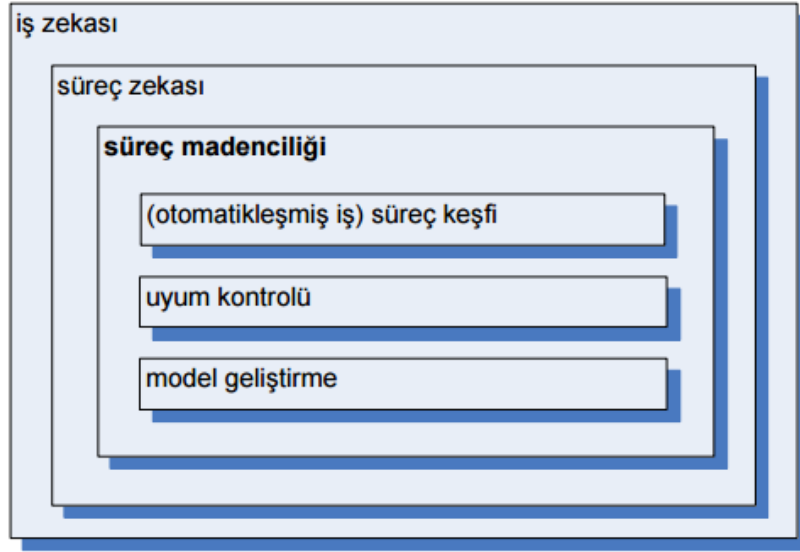
BÖLÜM 2

SÜREÇ MADENCİLİĞİ

Bilgi teknolojilerinde son yıllarda büyük verilerin yaygın bir şekilde kullanılması işletmelerde birtakım sorunları da beraberinde getirmektedir. Süreç madenciliği teknikleri kullanılarak süreçlerin yeniden analiz edilmesi, keşfi, modellenmesi ve süreçlerin iyileştirilmesi çalışmaları yapılarak verilerin etkin ve verimli olarak kullanılmasını sağlar. Böylelikle rekabetçi iş ortamında büyük işletmelerin süreç madenciliği tekniklerinin kullanılmasını zorunlu hale getirir.

Süreç madenciliği geleneksel iş zekası ve iş süreç takibi ortamlarını tamamlayan yeni bir teknolojidir. Olay kayıtlarından kullanışlı ve kolay olmayan bilgileri çıkaran, uyumluluğunu izleyen, performans analizini yapan, süreç tahmini yapan ve zenginleştiren bir uygulamadır. Daha çok ERP(Kurumsal Kaynak Planlaması), CRM (Müşteri İlişkileri Yönetimi) ve WFMS(İş akış yönetim sistemi) gibi alanlarda sıklıkla kullanılmaktadır [12,13].

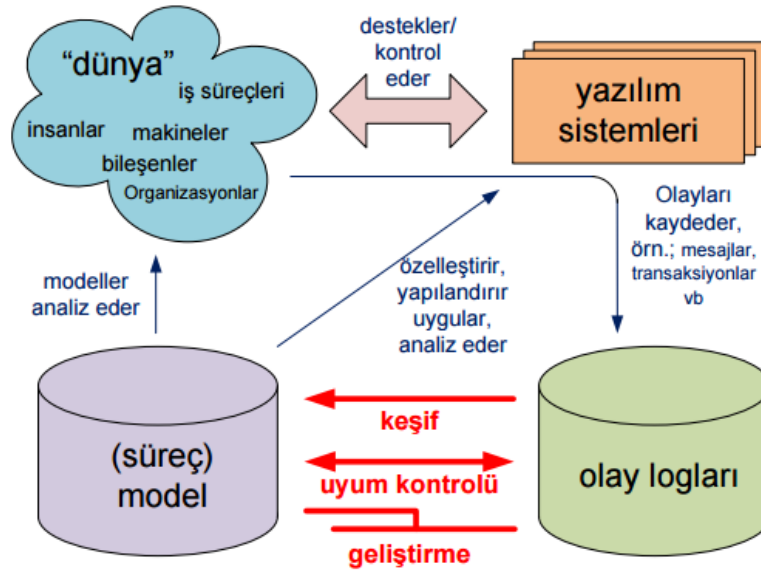
Süreç Madenciliğinin iş zekası ile ilişkisi Şekil 2.1’de gösterilmiştir [14,15]. Süreç madenciliği iş zekası altındaki tüm sistemleri destekler ve yararlı bilgiler sunar. Süreç zekası ise süreçleri belirlemek, analiz etmek ve geliştirmek için iş zekası yöntemlerini kullanır.



Şekil 2. 1 Süreç Madenciliğinin iş zekası ile ilişkisi[15]

2.1 Süreç Madenciliği Aşamaları

Süreç madenciliği süreçleri analiz eden, modelleyen, veri madenciliği yöntemlerini kullanan ve son yıllarda popüler olan yeni bir bilimsel disiplin alanıdır. Bu disiplin alanının amacı farklı algoritmaları kullanarak süreçleri belirlemek ve süreçlerin başarımlarını hesaplamaktır. Süreç madenciliğinin günümüzde popüler olmasının sebeplerinden biri de büyük veriler arasında istenen faydalı bilginin bulunmasıdır. Aynı zamanda iş dünyasında süreçlerin gelişmesi için katkı sağlar [14,15].



Şekil 2.2 Süreç madenciliğinin üç adımı [15]

Şekil 2.2’de süreç madenciliğinin temel işlev ve adımları gösterilmiştir [14,15]. Gerçek dünyadaki her bir süreç olay dizileri olarak yazılım sistemleri aracılığıyla olay kayıtlarında tutulur. Daha sonra bu olay kayıtlarındaki bilgiler süreç madenciliği işlemlerinden olan keşif, uygunluk kontrolü ve geliştirme ile modellenerek analiz edilir. Böylece anlamlı veriler kişilere ve sistemlere sunulur.

2.1.1 Süreçlerin Keşfi

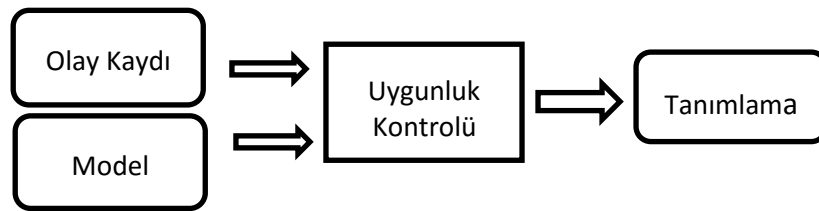
Şekil 2.3’de gösterildiği gibi süreçler olay kayıtlarından keşfedilir ve model oluşturulur. Bu süreç madenciliğinin en önemli aşamasıdır. Süreç keşfi yapılırken alfa, sezgisel, genetik başta olmak üzere çeşitli süreç madenciliği algoritmaları kullanılabilir.



Şekil 2.3 Süreç madenciliğinin keşif aşaması

2.1.2 Süreçlerin Uygunluk Kontrolü

Şekil 2.4’de süreçlerin uygunluk kontrolü aşaması gösterilmiştir. Bu aşamada süreçler izlenir, mevcut süreçler ile olay kayıtlarından elde edilen süreçlerin uyumluluğu kontrol edilir. Böylece olay kaydı ile model arasındaki farklılıkları bulmak, süreç içinde yerlerini tespit etmek ve şiddetini ölçmek için kullanılır

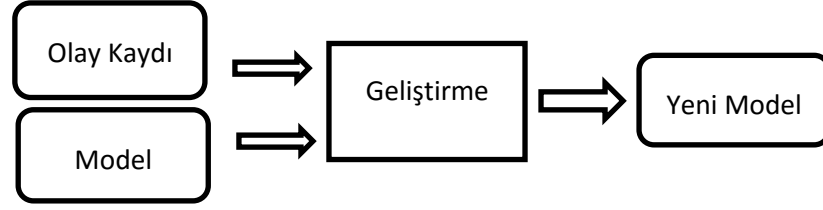


Şekil 2.4 Süreç madenciliğinin uygunluk kontrolü aşaması

2.1.3 Süreçlerin Geliştirilmesi

Şekil 2.5’de süreçlerin geliştirilmesi aşaması gösterilmiştir. Bu aşamanın amacı mevcut modelin eksik yönlerini tespit edip değiştirerek yeniden gelişmesini sağlamaktır. Örneğin, olay kayıtları incelenerek zaman açısından yüksek maliyetli olan aktiviteler

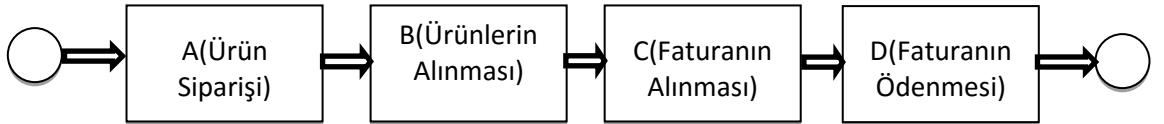
tespit edilir ve bu aktiviteler arasındaki tıkanıklıklar tespit edilerek yeniden modellenmesini sağlar.



Şekil 2.5 Süreç madenciliğinin geliştirme aşaması

2.2 Örnek Süreç Olay Kaydının İncelenmesi

Günümüzde olay kayıtlarının kaydedilmesi işletmelerdeki süreçlerin en önemli kısımlarından biridir. Kurumsal kuruluşlarda kullanılan hemen hemen her yazılımda olay kayıtları kaydedilerek incelenip gözlemleme işlemlerinin yapılmasına imkân sağlamaktadırlar. Kaydedilen bu olay kayıtlarından süreç hakkında analizler yapılabilir. Süreç madenciliği için örnek bir sipariş olay kaydı incelenerek sipariş ile ilgili detaylı bilgiler ve petri ağı incelenmiştir. Şekil 2.6’da örnek bir iş sürecini gösteren bir sipariş uygulaması verilmiştir.



Şekil 2.6 Örnek sipariş uygulaması [13]

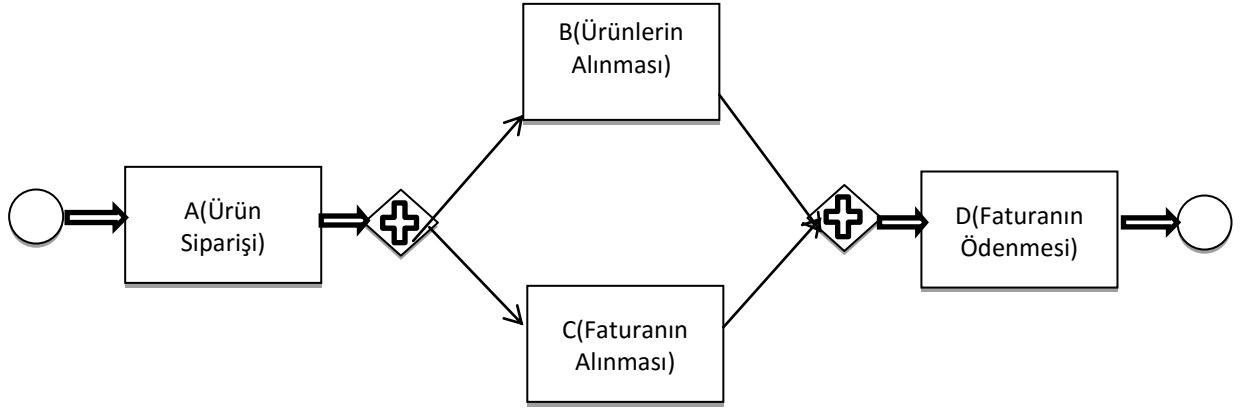
Burada sipariş alımı, fatura ödenmesi ve müşteriye teslimi belirli zamana göre gerçekleştirilmektedir. Bu iş süreci örneğinin olay kayıt tablosu Çizelge 2.1’de gösterilmiştir.

Çizelge 2.1 Sipariş örneğinin olay kayıt tablosu

Durum ID	Olay ID	Yapılma Zamanı	Aktivite	İşlemi Yapan Kişi
1	1000	01.01.2013	A(Ürün Siparişi)	Peter
1	1001	10.01.2013	B(Ürünlerin Alınması)	Micheal
1	1002	13.01.2013	C(Faturanın Alınması)	Frank
1	1003	20.01.2013	D(Faturanın Ödenmesi)	Tanja
2	1004	02.01.2013	A(Ürün Siparişi)	Peter
2	1005	03.02.2013	B(Ürünlerin Alınması)	Micheal
...	...	...	...	...

Süreç madenciliğinde olay kayıtları analiz edilerek sonuçlar Petri ağı başta olmak üzere BPMN (Business Process Model and Notation), EPC (Event-driven Process Chain) ve UML (Unified Modelling Language) kullanılarak ifade edilir. BPMN kurumsal işletmelerin iş süreçlerinin tasarımında ve optimizasyon çalışmalarında ortak bir dil oluşturulmasını amaçlayarak iş süreçlerini grafiksel olarak sunar. EPC süreçlerin paralel olarak çalışmasını sağlayan anlaşılması kolay olan bir kavramdır. AND, OR ve XOR mantıksal ifadeler ile iş akış süreçleri modellenir ve görselleştirilir. UML yazılım sistemlerini tanımlamak ve bu sistemler arasındaki ilişkileri görselleştirmek için

kullanılan diyagram çizme ve ilişkisel modelleme yapan nesne tabanlı standart bir dildir. Petri ağıları süreçleri analiz edilebilen, olay kayıtlarından ilgili aktivitelerin, geçişlerin ve durumların meydana gelmesini kontrol edebilen bir graf yapısıdır [16]. Şekil 2.7’de sipariş örneğinin petri ağı gösterilmiştir.



Şekil 2.7 Sipariş örneğinin petri ağı ile gösterimi [13]

2.3 Süreç Madenciliğinde Kullanılan Araçlar

Süreç madenciliğinde birçok algoritma kullanılmaktadır. Bunlardan en yaygın olarak kullanılanları Alfa, sezgisel, genetik ve fuzzy algoritmasıdır [17]. Süreç madenciliğinde en yaygın kullanılan platformlar ProM ve Disco’dur.

ProM Hollanda’da bulunan Eindhoven Teknoloji Üniversitesi tarafından geliştirilmiş süreç madenciliği keşif algoritmalarını tek bir platformda sunan açık kaynak kodlu bir sistemdir. ProM süreçleri keşfetmek, süreçlerin uygunluğunun ölçülmesi ve süreçlerinin iyileştirilmesini sağlayan Genetik, Fuzzy, Sezgisel ve Alfa gibi birçok süreç madenciliği algoritmalarının birbiriyle ilişkili olarak çalıştığını gösteren bir platformdur.

Disco 2009 yılında Dr. Anne Rozinat ve Dr. Christian Günther tarafından Fluxicon şirketi adına yapılmış bir sistemdir. Dico otomatik süreç keşfi yaparak süreçlerin haritasını görselleştiren ve böylece süreçlerin daha anlaşılır olmasını sağlayan platformdur. Çizelge 2.2’de süreç madenciliğinde kullanılan araçlar gösterilmiştir.

Çizelge2.2 Süreç madenciliğinde kullanılan araçlar [13]

Ürün Adı	Link
ARIS Process Performance Manager	www.softwareag.com
Celonis business intelligence	www.celonis.de
Disco	www.fluxicom.com
Genet/Petrify	www.lsi.upc.edu
Interstage Business Process Manager	www.fujitsu.com
QPR ProcessAnalyzer	www.gqr.com
ProM	www.processmining.org
ProcessGold	www.processgold.de
Rbminer/Dbminer	www.lsi.upc.edu
ReflectOne	www.pallas-athena.com
ServiceMosaic	Soc.cse.unsw.edu.au

SÜREÇ MADENCİLİĞİ KEŞİF ALGORİTMALARI

Süreç madenciliğinde birçok farklı algoritma kullanılmaktadır. Bu algoritmalar Alfa algoritması (Alpha algorithm), Çok Fazlı Yaklaşım algoritması (Multi Phase Miner), Sezgisel algoritma (Heuristic algorithm), Bulanık Yaklaşım (Fuzzy Miner) ve Genetik algoritmalarıdır. Tez çalışması kapsamında alfa, sezgisel ve genetik algoritmalar incelenmiştir.

3.1 Alfa Algoritması

Alfa algoritmasının amacı olay kayıtlarından bir petri ağı oluşturmaktır. Alfa algoritması adımlarının iyi anlaşılması için aşağıdaki teoremlerden bahsedilmiştir.

Teorem 3.1

Doğrusal ilişki ($>$): $x > y$ olduğunda iş akış ağında **y** aktivitesi **x** aktivitesini izler.

Teorem 3.2

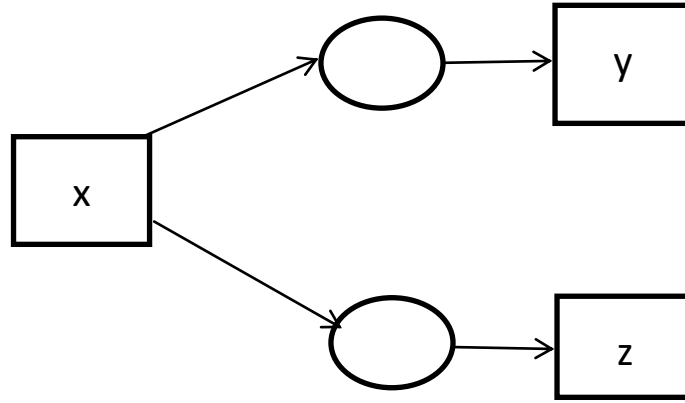
Nedensellik ($\rightarrow$): Eğer $x > y$ aktiviteleri arasında doğrusal ilişki var ama $y > x$ aktiviteleri arasında doğrusal ilişki yoksa bu iş akış aktiviteleri arasında nedensel ilişki vardır.



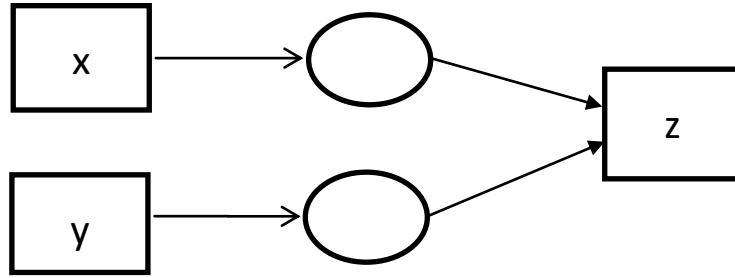
Şekil 3.1 x ve y aktiviteleri arasında nedensel ilişki gösterimi($x \rightarrow y$)

Teorem 3.3

Paralellik (||):Eğer $y > z$ aktiviteleri ve $z > y$ aktiviteleri her ikisi arasında doğrusal ilişki varsa $y \parallel z$ aktiviteleri arasında paralel ilişki vardır. Şekil 3.2'de $x \rightarrow y$ aktiviteleri ve $x \rightarrow z$ aktiviteleri arasında nedensel ilişkiler olduğundan $y \parallel z$ aktiviteleri arasında paralel ilişki vardır. Şekil 3.3'te ise $x \rightarrow z$ aktiviteleri ve $y \rightarrow z$ aktiviteleri arasında nedensel ilişkiler olduğundan $x \parallel y$ aktiviteleri arasında paralel ilişki vardır.



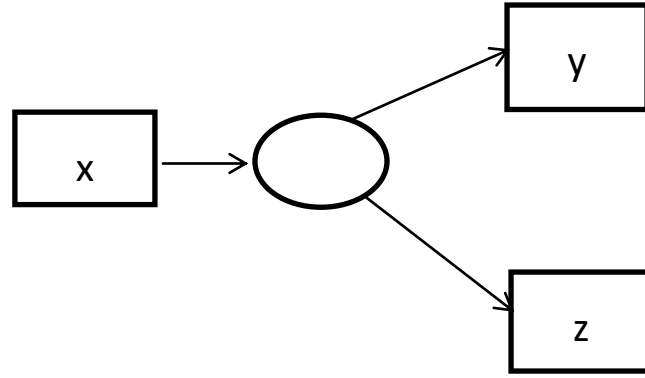
Şekil 3.2 y ve z aktiviteleri arasındaki paralellik gösterimi($y \parallel z$)



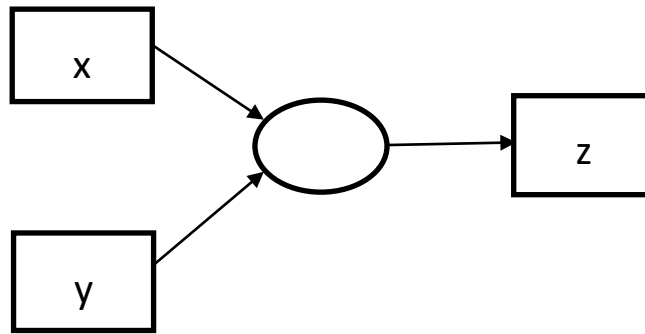
Şekil 3.3 x ve y aktiviteleri arasındaki paralellik gösterimi($x \parallel y$)

Teorem 3.4

Seçme (#): Eğer $y > z$ aktiviteleri ve $z > y$ aktiviteleri her ikisi arasında doğrusal ilişki yoksa $y \# z$ aktiviteleri arasında seçme ilişkisi vardır. Şekil 3.4'de $x \rightarrow y$ aktiviteleri arasında ve $x \rightarrow z$ aktiviteleri arasında nedensel ilişkiler olduğundan $y \# z$ aktiviteleri arasında seçme ilişkisi vardır. Şekil 3.5'de ise $x \rightarrow z$ aktiviteleri arasında ve $y \rightarrow z$ aktiviteleri arasında nedensel ilişkiler olduğundan $x \# y$ aktiviteleri arasında seçme ilişkisi vardır.



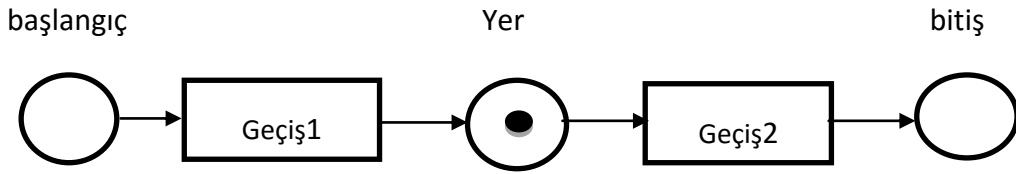
Şekil 3.4 y ve z aktiviteleri arasında seçme işleminin gösterimi($y \# z$)



Şekil 3.5 x ve y aktiviteleri arasında seçme işleminin gösterimi($x \# y$)

3.1.1 Alfa Algoritması Adımları

Alfa algoritmasının amacı olay kayıtları kullanılarak petri ağı oluşturmaktır. Petri ağı yerler (places) ve geçişler (transitions) olmak üzere iki özellikten oluşur. Şekil 3.6’da gösterilen Petri ağında yuvarlak şekiller yerleri gösterirken; kare şekilleri ise geçişleri gösterir. Aynı zamanda bir başlangıç yeri ve bir bitiş yeri bulunur. Yer, halka ve geçişler (place-token-transition) petri ağında gösterilmiştir [18,19]. Başlangıç yerinden (start place) diğer aktiviteye geçişte halka eklenir. Bir sonraki aktiviteye geçişte ise halka kaldırılır ve diğer aktiviteye aktarılır. En son bitiş yeri (end place) gösterilir.



Şekil 3.6 Yer, geçiş ve halka gösterimi

Alfa algoritması adımları $(\alpha(W) = (P_W, T_W, F_W))$ aşağıda verilmektedir [20,21]. Bu algorithmada gösterilen P_W oluşan petri ağının yerlerini, T_W geçişlerini, F_W ise giriş ağından çıkış ağına kadar uzanan tüm yer ve geçişlerin birleştirilmiş halini yani oluşturulan petri ağını gösterir [20].

Adım 1: T_W olay kayıtlarının tüm geçişleri oluşturulur.

$$T_W = \{t \in T \mid \exists \sigma \in W \ t \in \sigma\},$$

Adım 2: T_I kaynak düğümdeki giriş yerleri oluşturulur.

$$T_I = \{t \in T \mid \exists \sigma \in W \ t = \text{first}(\sigma)\},$$

Adım 3: T_O alıcının çıkış geçişleri oluşturulur.

$$T_O = \{t \in T \mid \exists \sigma \in W \ t = \text{last}(\sigma)\},$$

Adım 4: Petri ağındaki ağlarında keşfedilen yerler tanımlanır (X_W ve Y_W).

$$X_W = \{(A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow W^b \wedge \forall_{a1,a2 \in A} a1 \# W^{a2} \wedge \forall_{b1,b2 \in B} b1 \# W^{b2}\},$$

Adım 5: Petri ağlarında keşfedilen yerler tanımlanır (X_W ve Y_W).

$$Y_W = \{(A,B) \in X_W \mid \forall_{(A',B') \in X_W} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\},$$

Adım 6: Her giriş ve çıkış ağlarının ayrı ayrı yerleri oluşturulur.

$$P_W = \{p(A, B) \mid (A,B) \in Y_W\} \cup \{i_W, o_W\},$$

Adım 7: Adım 6 da elde edilen yerler birleştirilerek petri ağı oluşturulmaya başlanır.

$$F_W = \{(a, p(A, B)) \mid (A,B) \in Y_W \wedge a \in A\} \cup \{(p(A, B), b) \mid (A, B) \in Y_W \wedge b \in B\} \cup \{(i_W, t) \mid t \in T_I\} \cup \{(t, o_W) \mid t \in T_O\}$$

3.1.2 Alfa Algoritması için Örnek Uygulama

Alfa algoritması için örnek olarak L1 olay kayıtları $[(a,b,c,d)^3, (a,c,b,d)^2, (a,e,d)]$ olsun [4]. L1 abcd, acbd, aed olmak üzere 3 farklı durumdan oluşur. Alfa algoritmasında kullanılan doğrusal ilişki, nedensellik, paralel ve seçim ilişkileri aşağıda sırasıyla gösterilmiştir.

1) Doğrusal izlemeyi bulabilmek için tüm aktiviteler kendi aralarında ikişerli gruplar halinde yazılır.

ab, bc, cd, ac, cb, bd, ae, ed

2) Nedensellik ilişki doğrusal ilişkiler kullanılarak ($x > y$ ilişkisi olacak ama $y > x$ ilişkisi olmayacak) bulunur. Burada b c'yi ve c'de b'yi takip ettiğinden bu aktivitelerin nedensellik ilişkisi bulunmaz. Bunların dışındaki aktiviteler arasındaki ilişkiler nedensel ilişki olarak kabul edilir.

ab, cd, ac, bd, ae, ed

3) Paralellik ilişkisini bulabilmek için $x > y$ ve $y > x$ doğrusal ilişkiyi sağlayan aktiviteler alınır. Bu durumda b'den c'ye c'den b'ye ilişki olduğundan bu aktivite grupları paralel olarak değerlendirilir.

bc, cb

4) Seçme ilişkisini bulabilmek için $x > y$ ve $y > x$ doğrusal ilişki olmayan aktivite grupları alınır. Bu durumda tüm aktivitelerin en başta kendi kendini seçme olanağı alınır. Daha sonra aralarında doğrusal ilişki olmayan tüm aktivite grupları arasında seçme ilişkisi bulunur.

aa, bb, cc, dd, ee, ad, be, ce, da, eb, ec

L1 olay kaydının doğrusal ilişkisi, nedenselliği, aralarındaki seçim ve paralellik gösterimleri sırasıyla aşağıdaki gibi belirtilir:

$>L1 = \{ (a, b), (a, c), (a, e), (b, c), (c, b), (b, d), (c, d), (e, d) \}$

$\rightarrow L1 = \{ (a, b), (a, c), (a, e), (b, d), (c, d), (e, d) \}$

$\#L1 = \{ (a, a), (a, d), (b, b), (b, e), (c, c), (c, e), (d, a), (d, d), (e, b), (e, c), (e, e) \}$

$||L1 = \{ (b, c), (c, b) \}$

Yukarıdaki $\rightarrow L1$, $\#L1$, $||L1$ ve $\leftarrow L1$ ilişkilere göre Çizelge 3.1'deki olay izleri (footprint) oluşturulur [4].

Çizelge 3.1 L1 olay kaydının izleri

	a	b	c	d	e
a	$\#L1$	$\rightarrow L1$	$\rightarrow L1$	$\#L1$	$\rightarrow L1$
b	$\leftarrow L1$	$\#L1$	$ L1$	$\rightarrow L1$	$\#L1$
c	$\leftarrow L1$	$ L1$	$\#L1$	$\rightarrow L1$	$\#L1$
d	$\#L1$	$\leftarrow L1$	$\leftarrow L1$	$\#L1$	$\leftarrow L1$
e	$\leftarrow L1$	$\#L1$	$\#L1$	$\rightarrow L1$	$\#L1$

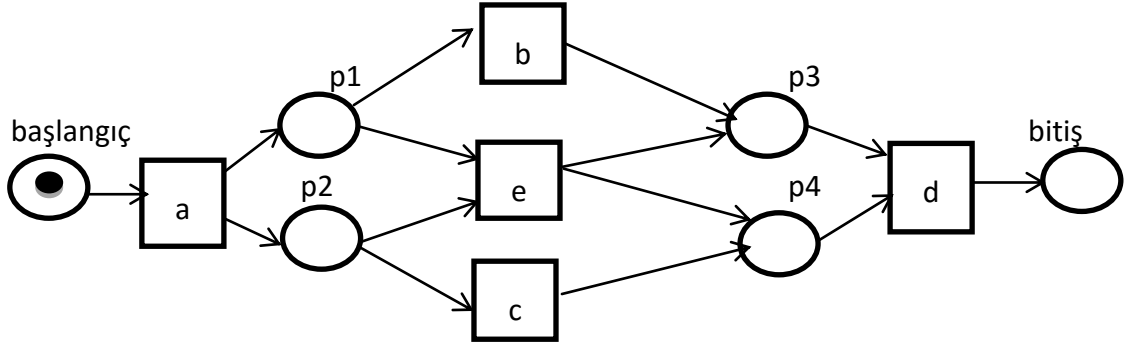
Alfa algoritmasında aktiviteler arasında ilişkileri gösteren olay izleri kullanılarak petri ağının yerleri bulunur. X_{L1} oluşturulurken önce ikili elemanlar arasındaki nedensel ilişkiler alınır ve aralarına yerler eklenir. Daha sonra paralel ilişkiler dışındaki tüm nedensel ilişki bulunan çoklu elemanlar arasına yerler eklenir.

$$X_{L1} = \{(\{a\},\{b\}), (\{a\},\{c\}), (\{a\},\{e\}), (\{a\},\{b,e\}), (\{a\},\{c,e\}), (\{b\},\{d\}), (\{c\},\{d\}), (\{e\},\{d\}), (\{b,e\},\{d\}), (\{c,e\},\{d\})\}$$

X_{L1} 'den Y_{L1} 'ye geçerken X_{L1} 'de bulunan ikili aktivite grupları atılır. Üçlü aktivite grupları ise Y_{L1} kümesine aktarılır. Çünkü X_{L1} kümesinde ikili gruplar arasına yerler daha önce eklendiği için Y_{L1} üçlü gruplar arasına tekrar yerler ekleneneğinden X_{L1} 'deki yerler alınmaz. Böylece tekrar yer ekleme problemi önlenmiş olur.

$$Y_{L1} = \{(\{a\},\{b,e\}), (\{a\},\{c,e\}), (\{b,e\},\{d\}), (\{c,e\},\{d\})\}$$

Yukarıdaki örnek verilen L1 olay kaydını alfa algoritmasına göre petri ağı Şekil 3.7'deki gibi elde edilir.



Şekil 3.7 L1 olay kaydının petri ağı

3.2 Sezgisel Algoritma

Sezgisel algoritma bir problemin çözümünün yüzde yüz doğru olup olmadığına bakmaksızın en iyi çözümü arayan algoritmadır. Problemi çözerken iki yaklaşım sunar:

Ya hızlı bir şekilde sonuca ulaşır ya da makul bir zamanda daha iyi bir sonuç almayı hedefler.

Sezgisel algoritmadaki amaç olay kayıtları kullanarak en iyi çözümü sağlayan bir petri ağı oluşturmaktır. Sezgisel algoritma alfa algoritmasının geliştirilmiş halidir. Alfa algoritmasının dezavantajlı yanları olan verideki gürültü ve tamamlanmayan süreçleri geliştirerek alfa algoritmasına göre avantaj sağlamaktadır.

3.2.1 Sezgisel Algoritma Adımları

Sezgisel algoritma olay kayıtlarındaki geçişleri sıraya koyar ve frekansları temel alarak az frekansa sahip olan geçişleri kaldırır. Sezgisel algoritmanın adımları aşağıdaki gibidir [22].

Adım 1: Olay kayıtlarından tüm geçişler alınır

$$T = \{t \mid \exists_{\sigma \in W^+} [t \in \sigma]\},$$

Adım 2: Tek döngü uzunluğu (length-one loop) için geçişler hesaplanır.

$$C_1 = \{(a, a) \in T \times T \mid a \Rightarrow W^a \geq \sigma L1L\} \text{ (tek uzunluklu döngü)},$$

Adım 3: İki döngü uzunluğu (length-two loop) için geçişler hesaplanır.

$$C_2 = \{(a, b) \in T \times T \mid (a, a) \notin C_1 \wedge (b, b) \notin C_1 \wedge a \Rightarrow 2 W^b \geq \sigma L2L\} \text{ (çift uzunluklu döngü)},$$

Adım 4: En güçlü çıkış değeri bulunur.

$$C_{out} = \{(a, b) \in T \times T \mid b \neq \text{End} \wedge a \neq b \wedge \forall_{y \in T} [a \Rightarrow W^{b \geq a} \Rightarrow W^y]\},$$

Adım 5: En güçlü giriş değeri bulunur.

$$C_{in} = \{(a, b) \in T \times T \mid a \neq \text{Start} \wedge a \neq b \wedge \forall_{x \in T} [a \Rightarrow W^{b \geq x} \Rightarrow W^b]\},$$

Adım 6: İki döngü uzunluğu için en zayıf olan çıkış bağlantıları bulunur.

$$C'_{out} = \{(a, x) \in C_{out} \mid (a \Rightarrow W^x) < \sigma_a \wedge \exists_{(y,b) \in C_{out}} [(a, b) \in C_2 \wedge ((b \Rightarrow W^y) - (a \Rightarrow W^x)) > \sigma_r]\}$$

Adım 7: Zayıf çıkış bağlantıları atılır.

$$C_{out} = C_{out} - C'_{out} ,$$

Adım 8: İki döngü uzunluğu için en zayıf olan giriş bağlantıları bulunur.

$$C'_{in} = \{(x, a) \in C_{in} \mid (x \Rightarrow W^a) < \sigma_a \wedge \exists_{(y,b) \in C_{in}} [(a, b) \in C_2 \wedge ((y \Rightarrow W^b) - (x \Rightarrow W^a)) > \sigma_r]\}$$

Adım 9: Zayıf giriş bağlantıları atılır.

$$C_{in} = C_{in} - C'_{in}.$$

Adım 10: Çıkış geçişleri hesaplanması eşik değere göre belirlenir.

$$C''_{out} = \{(a, b) \in T \times T \mid a \Rightarrow W^b \geq \sigma_a \vee \exists (a,c) \in C_{out} [((a \Rightarrow W^c) - (a \Rightarrow W^b)) < \sigma_r]\},$$

Adım 11: Giriş geçişleri eşik değere göre belirlenir.

$$C''_{in} = \{(b, a) \in T \times T \mid (b \Rightarrow W^a) \geq \sigma_a \vee \exists (b,c) \in C_{in} [((b \Rightarrow W^c) - (b \Rightarrow W^a) < \sigma_r)]\},$$

Sonuç olarak bağlı graf oluşturulur. ($DG = C_1 \cup C_2 \cup C''_{out} \cup C''_{in}$).

3.2.2 Sezgisel Algoritma için Örnek Uygulama

Örnek olarak L2 olay kaydı $[(a,e)^5, (a,b,c,e)^{10}, (a,c,b,e)^{10}, (a,b,e)^1, (a,c,e)^1, (a,d,e)^{10}, (a,d,d,e)^2, (a,d,d,d,e)^1]$ olsun [17].

Sezgisel algoritmada alfa algoritmasından farklı olarak olay kayıtları içerisindeki aktivite grupları arasında frekans değerleri hesaplanır. Kısacası alfa algoritmasındaki her olay kaydı bir kez alınırken; sezgisel algoritmada ise olay sayıları kullanılarak frekans hesaplanılır. Frekans hesabı ise aralarında doğrusal ilişki olan aktivite gruplarının toplamı alınarak hesaplanılır. Çizelge 3.2’de L2 olay kaydının frekans değerleri gösterilmiştir.

Çizelge 3.2 L2 olay kaydının izleri

$ >L $	a	b	c	d	e
a	0	11	11	13	5
b	0	0	10	0	11
c	0	10	0	0	11
d	0	0	0	4	13
e	0	0	0	0	0

Çizelge 3.2’de gösterilen frekans değerlerinin bağlı ölçüm(dependency measure) göre yapılmış hali Çizelge 3.3’de gösterilmiştir. Bağlı ölçüm değeri -1 ile +1 arasında değerler alır. Aktiviteler arasındaki ters ilişki grupları negatif olarak alınır. Bağlı ölçüm Eşitlik 3.1 ve Eşitlik 3.2’deki gibi hesaplanır.

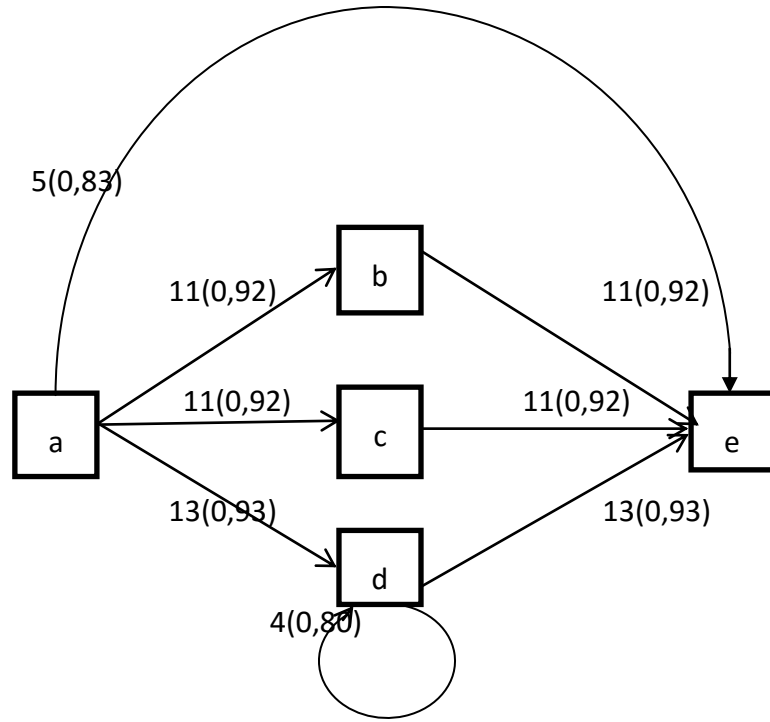
$$|a \rightarrow L_b| = \frac{|a > L_b| - |b > L_a|}{|a > L_b| + |b > L_a| + 1} \quad a \neq b \quad (3.1)$$

$$|a \rightarrow L_b| = \frac{|a > L_a|}{|a > L_a| + 1} \quad a = b \quad (3.2)$$

Çizelge 3.3- L2 olay kaydının nedensellik ilişkisinin hesaplanması

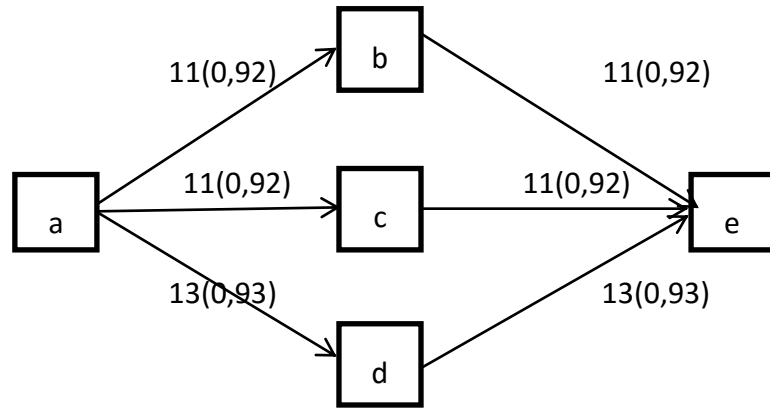
$I \rightarrow L \mid$	a	b	c	d	e
a	0	0.92	0.92	0.93	0.83
b	-0.92	0	0	0	0.92
c	-0.92	0	0	0	0.92
d	-0.93	0	0	0.80	0.93
e	-0.83	-0.92	-0.92	-0.93	0

Çizelge 3.2’ye göre çeşitli eşik değerler verilerek düşük frekansa sahip olan aktivite grupları atılır. Örnek olarak Şekil 3.8’de eşik değeri 2 verilmiştir ve frekansı 2’den büyük olan aktivite grupları kalır.



Şekil 3.8 Frekans değeri 2'den büyük olan aktivite grupları

Şekil 3.9'da eşik değeri 5 verilmiştir ve frekansı 5'den büyük olan aktivite grupları kalır küçük olanlar ise kaldırılır.

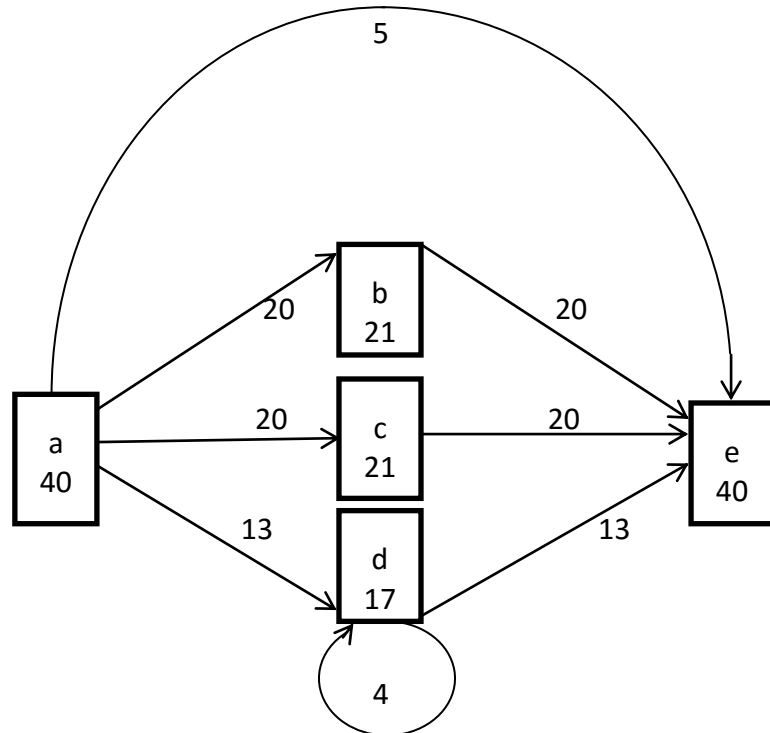


Şekil 3.9 Frekans değeri 5'den büyük olan aktivite grupları

Şekil 3.10'da görüldüğü gibi tüm aktivitelerin ve olay izlerinin bağlantı frekansları Çizelge 3.2 'ye göre hesaplanır. a'dan d'ye geçiş frekansı 13, a'dan e'ye geçiş frekansı 5, d'den e'ye geçiş frekansı 13 ve d'den d'ye geçiş frekansı 4 olarak bulunur. Ancak b ve c

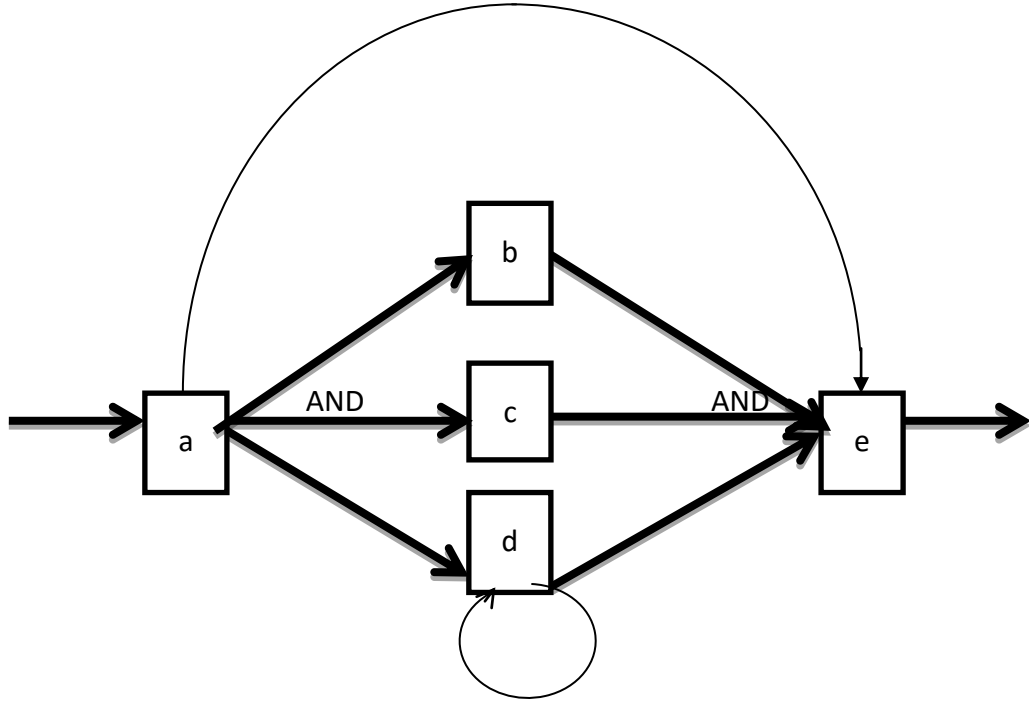
aktiviteleri paralel olduğundan geçiş frekansları farklı olarak hesaplanır. $b > c$ doğrusal ilişki frekansı 10 ve $c > b$ doğrusal ilişki frekansı 10 olduğundan a'dan b'ye ve a'dan c'ye geçiş frekans değerleri 20 ($10+10$) olarak bulunur.

Çizelge 3.2 'ye göre O(a) çıkış bağlantı (output binding) değerleri hesaplanırken $a > b$ doğrusal ilişki frekansı 11, $a > c$ doğrusal ilişki frekansı 11, $a > d$ doğrusal ilişki frekansı 13 ve $a > e$ doğrusal ilişki frekansı 5 olduğundan toplamda 40 dir ($11+11+13+5$). O(b) çıkış bağlantı değerleri hesaplanırken $b > c$ doğrusal ilişki frekansı 10, $b > e$ doğrusal ilişki frekansı 11 olduğundan toplamda 21 dir ($10+11$). O(c) çıkış bağlantı değerleri hesaplanırken $c > b$ doğrusal ilişki frekansı 10, $c > e$ doğrusal ilişki frekansı 11 olduğundan toplamda 21 dir ($10+11$). O(d) çıkış bağlantı değerleri hesaplanırken $d > d$ doğrusal ilişki frekansı 4, $d > e$ doğrusal ilişki frekansı 13 olduğundan toplamda 17 dir ($4+13$).



Şekil 3.10 Sezgisel algoritmaya göre oluşan petri ağı

Şekil 3.11'deki gibi ince bağlantılar zayıf frekansları gösterirken; kalın renkli bağlantılar ise yüksek frekanslı yolları gösterir. İnce bağlantılar yol haritası görselleştirirken önemsiz olduğundan kaldırılır.



Şekil 3.11 Sezgisel algoritma ile elde edilen yolların görselleştirilmesi

3.3 Genetik Algoritma

Genetik algoritma arama ve optimizasyon problemlerinin çözümünde modelleme yapabilmek için kullanılan biyolojik süreçlerin esas alındığı yöntemlerdendir. Genetik algoritmalar geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılır [23]. Genetik algoritmalar rastgele arama metodu olduğu için problemin optimum çözümü için tek bir çözüm aramak yerine birden fazla çözüm kümesi üzerinde çalışır. Bu nedenle problem çözümünde sonuçlar her zaman en iyi olmaz. Genetik algoritmanın tercih edilmesinin nedeni, genetik algoritmanın problemin doğasıyla ilgili herhangi bir bilgiye ihtiyaç duymamasıdır.

Genetik algoritma ilk olarak 1950’lerde ortaya çıkmıştır. 1960’larda Holland genetik algoritma ve optimizasyonla ilgili çalışmalara başlamıştır. Holland’ın amacı genetik algoritma kullanarak en iyi çözümü bulabilmektir [24]. Böylece evrim sürecini bilgisayar sistemlerine adapte etmeye çalışmıştır. 1975 yılında “Doğal ve Yapay Sistemlerde Uyumlaştırma” kitabında yapay problemlerde genetik algoritmanın kullanılabileceğini söylemiştir. Böylece Holland genetik algoritmanın öncüsü olarak kabul edilmektedir.

Koza 1992 yılında evrimsel süreci kullanarak genetik algoritmayı programlamayı ileri bir seviyeye getirmiştir [25].

1967 yılında Bagley geliştirdiği oyun programında genetik algoritma kullanarak optimizasyon problemine katkı sağlamıştır. Aynı dönemde Rosenberg biyolojik etkenleri aktarmıştır. Goldberg ise teorik olan genetik algoritmayı pratiğe dönüştürmüştür ve genetik algoritmanın farklı konularda kullanılabileceğini ispatlamıştır [26].

Genetik algoritma optimizasyon başta olmak üzere otomatik programlama ve bilgi sistemleri, mekanik öğrenme, finans, çizelgeleme problemleri, tesis yerleşim problemleri, gezgin satıcı problemleri, araç rotalama problemleri gibi farklı alanlarda yaygın bir şekilde kullanılmaktadır [27].

3.3.1 Genetik Algoritmanın Genel Tanımları

Genetik algortmada kullanılan genel terimler aşağıda anlatılmıştır [28].

Popülasyon (Topluluk): Problemde çözümü sağlayan tüm bireyleri içerir.

Kromozom: Herbir birey kromozom olarak ifade edilir.

Gen: Kromozom içindeki anlamlı en küçük bilgidir.

Uygunluk (fitness): Bir bireyin çözüme ne kadar yakın olduğunu ifade eder.

Uygunluk Fonksiyonu: Bir problemin optimal çözümü için oluşturulan fonksiyondur. Uygunluğu yüksek olan bireyin, yeni nesle aktarılma ihtimali de daha yüksek olacaktır.

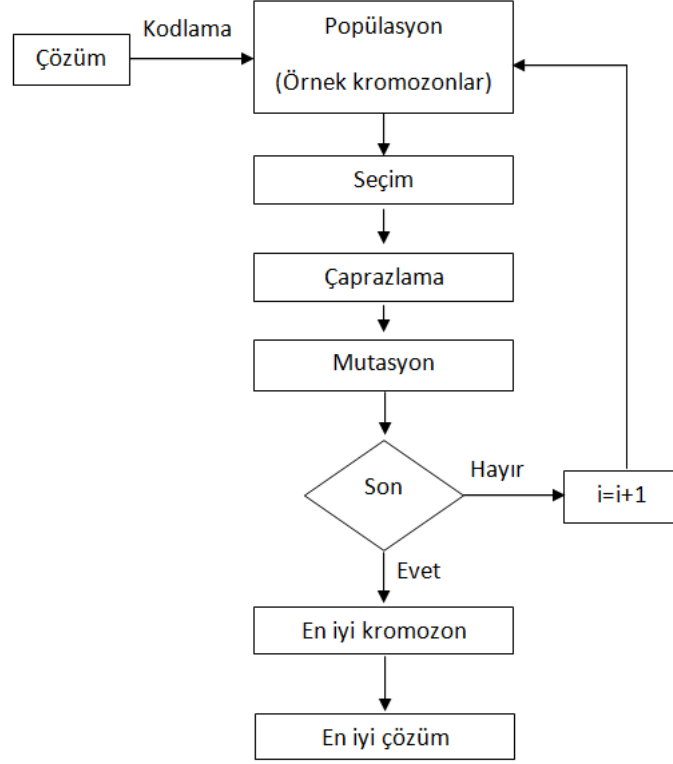
Çaprazlama (crossover): Problem uzayında iki en uygun bireyin kromozomlarını değiştirerek iki yeni yavru oluşmasını sağlar.

Mutasyon (mutation): Bir bireyde rastgele bir geni değiştirme işlemidir.

Seçme (selection): Çaprazlama için yeni bireylerin seçilme işlemidir.

3.3.2 Genetik Algoritmanın Genel Adımları

Genetik algoritma adımları aşağıda Şekil 3.12 'de gösterilmiştir [29].



Şekil 3.12 Genetik algoritmanın adımları [29]

Adım 1 Birey sayısı, mutasyon ve çaprazlama oranını belirler.

Adım 2 Rastgele popülasyon oluşturulur.

Adım 3 Yeni popülasyon oluşuncaya kadar Adım 4'ten Adım 7'ye kadar olan adımları tekrarla.

Adım 4 Seçme işlemi yapılır.

Adım 5 Çaprazlama işlemi yapılır.

Adım 6 Mutasyon işlemi yapılır.

Adım 7 Yeni yavru birey oluşturulur.

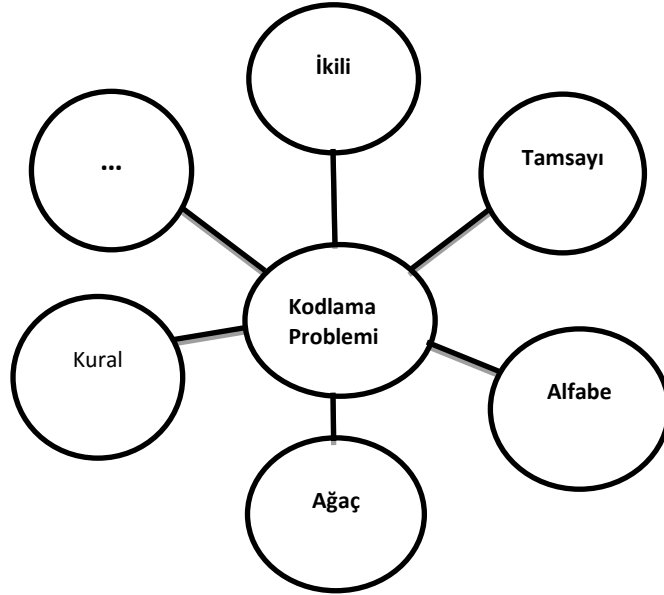
Adım 8 Problemi çözen en iyi bireyler oluşturulur.

3.3.3 Genetik Algoritmanın İşlemleri

Genetik algoritma uygulanırken kodlama, seçme, çaprazlama ve mutasyon işlemleri yapılır.

3.3.3.1 Kodlama

Şekil 3.13’de genetik algorithmada kullanılan kodlama çeşitleri verilmiştir. Bu kodlama çeşitlerinden en yaygın kullanılanı ikili kodlama (binary encoding) türüdür [28].



Şekil 3.13 Kodlama çeşitleri[28]

İkili kodlamada her gen 0 ve 1 ‘e karşılık gelir. Burada Kromozom 1 ve Kromozom 2 olmak üzere iki farklı kromozomun ikili kodlanmış hali aşağıda gösterilmiştir.

Kromozom1 1101100100110110

Kromozom2 1101111000011110

3.3.3.2 Seçme

Rastgele oluşturulan popülasyondan uygunluk değerlerine göre çaprazlama işlemi için yeni bireylerin oluşturulmasıdır. Birçok seçim türü kullanılmaktadır. Bunlara örnek verecek olursak rulet çemberi, rank seçimi, rastgele seçim ve turnuva seçimidir.

Rulet çemberi: Genetik algorithmada seçim yöntemi olarak kullanılan en sık yöntemdir. Bireylerin uygunluğuna göre uygunluk fonksiyonu yüksek olan bireylerin yeni popülasyona dahil edilme ihtimali daha fazla olur. Aşağıda verilen bir rulet seçim

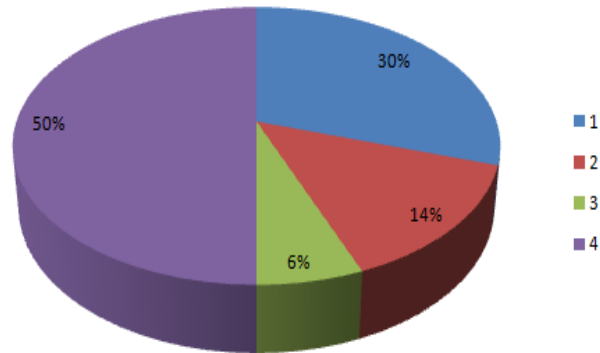
örneğinde her bireyin uygunluk fonksiyonuna göre yüzdelik seçilme oranları hesaplanarak Şekil 3.14'de gösterilmiştir.

Birey 1: $f(x) = 45 \rightarrow \%30$

Birey 2: $f(x) = 21 \rightarrow \%14$

Birey 3: $f(x) = 9 \rightarrow \%6$

Birey 4: $f(x) = 75 \rightarrow \%50$



Şekil 3.14 Rulet çemberi

Rank seçimi: Rulet seçim yönteminde en yüksek uygunluk derecesine sahip bireyin sürekli seçilmesi bazen sıkıntı oluşturabilir buna alternatif olarak rank seçim yöntemi kullanılabilir. n adet popülasyonda uygunluk derecesi en yüksek olan bireylere n değeri verilir. Sırasıyla uygunluk derecesi en küçük olan bireye kadar n değeri azaltılarak ve uygunluk derecesi en küçük olan birey 1 değerini alır. Aşağıda bir rank seçim örneği verilmiştir:

Birey 1: $f(x) = 2 \rightarrow 1$

Birey 2: $f(x) = 5 \rightarrow 2$

Birey 3: $f(x) = 7 \rightarrow 3$

Birey 4: $f(x) = 11 \rightarrow 4$

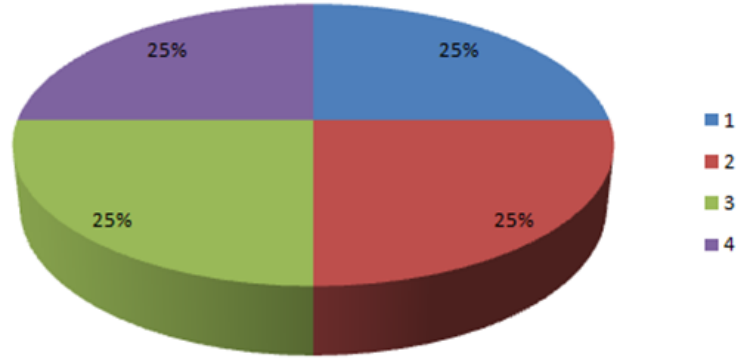
Rastgele seçim: Popülasyondaki bireylerin uygunluk değerine bakılmaksızın çaprazlama işlemi için rastgele bireyler seçilir.

Birey 1: $f(x) = 9$

Birey 2: $f(x) = 11$

Birey 3: $f(x) = 23$

Birey 4: $f(x) = 35$



Şekil 3.15 Rastgele seçme işlemine göre bireylerin seçim oranları

Turnuva seçimi: Rastgele seçilen iki birey kendi arasında karşılaştırılarak uygunluk derecesi yüksek olan birey alınır ve bu işlem popülasyondaki kromozom sayısı kadar tekrarlanır.

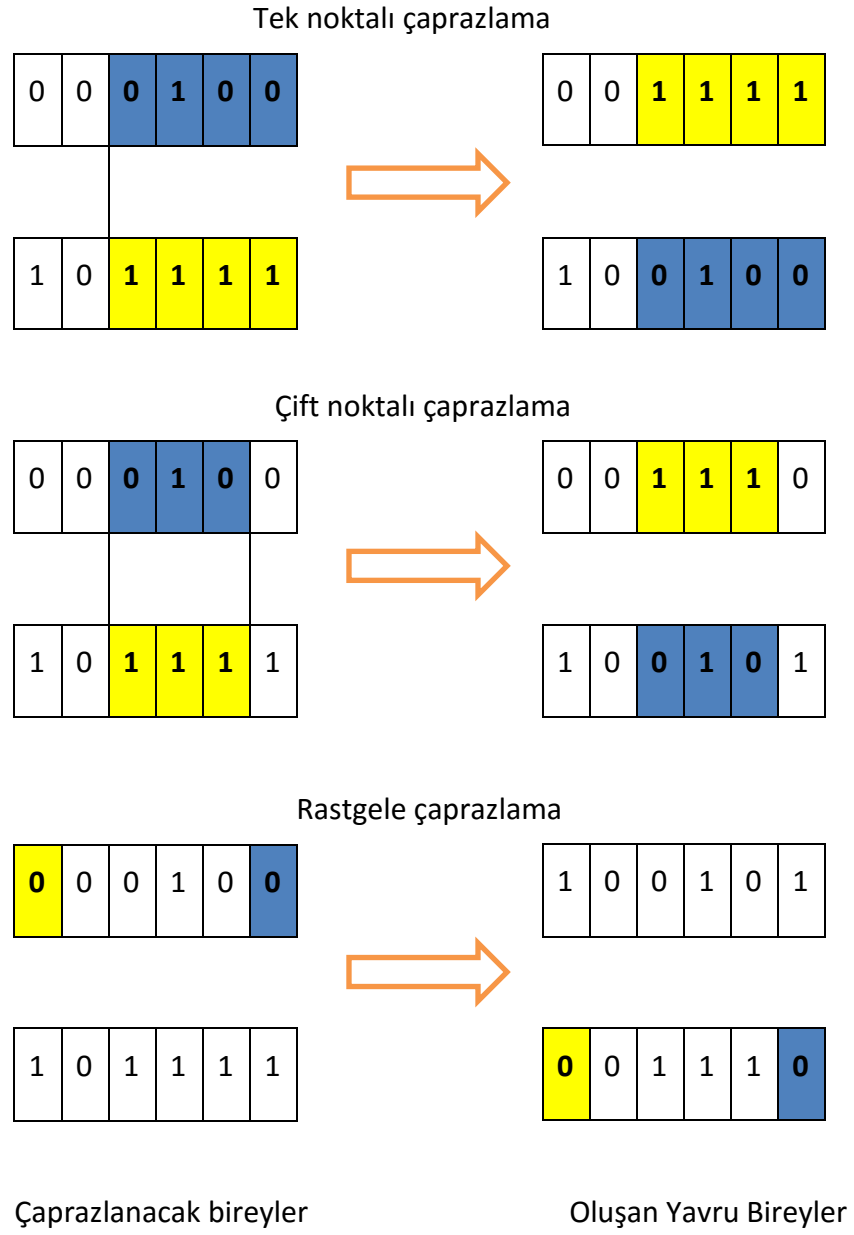
3.3.3.3 Çaprazlama

Çaprazlama metotları Şekil 3.16'da gösterilmiştir [30]. Genetik algortımanın en önemli adımını oluşturur.

Tek noktalı çaprazlama (one point crossover): İki birey arasında çaprazlama yapılırken iki birey içinde rastgele tek bir nokta seçilir. Seçilen kısım aynı uzunlukta olur ve bu seçilen kısımlar kendi aralarında değiştirilir.

İki noktalı çaprazlama (two point crossover): İki birey arasında çaprazlama yapılırken iki birey içinde rastgele iki nokta seçilir. Seçilen kısım aynı uzunlukta olur ve bu seçilen kısımlar kendi aralarında değiştirilir.

Uniform çaprazlama : İki birey arasında çaprazlama yapılırken iki birey içinde rastgele noktalar seçilir. Seçilen kısımlar kendi aralarında değiştirilir.



Şekil 3.16 Çaprazlama metotları [30]

3.3.3.4 Mutasyon

Popülasyondaki herhangi bir bireyin rastgele bir geninin değiştirilme işlemidir. Amacı popülasyondaki çeşitliliği arttırmaktır.

Farklı mutasyon türleri vardır tersleme (inversion), ekleme (insertion), çıkarma (displacement) ve yerdeğiřtirme (reciprocal exchange) gibi en sık kullanılan türü ise tersleme yöntemidir Çizelge 3.4'de tersleme yöntemi ile ilgili bir örnek verilmiştir.

Çizelge 3.4 Bireylerin mutasyon işlemi

Asıl Yavru1	1101111000011110
Asıl Yavru2	1101100100110110
Mutasyonlu Yavru1	1100111000011110
Mutasyonlu Yavru2	1101101100110100

3.3.4 Genetik Algoritma ile Süreç Keşfi

Genetik algoritma kullanarak süreç keşif işlemi örnek bir olay kaydı kullanarak açıklanmıştır. 18 olay, 4 durum ve 8 farklı aktivite olan örnek olay kaydı Çizelge 3.5'deki gibi gösterilmiştir. Burada herbir olayın 4 farklı özelliği vardır [18,19]. Bunlar durum bilgisini gösteren durum ID'si, aktiviteleri gösteren aktivite ID'si, işlemin kim tarafından gerçekleştirildiği ve hangi zamanda yapıldığıdır. Genetik algoritma kullanılarak süreç keşfi ve başarımlı oranı hesaplanması istenmektedir. Bunlar yapılırken Çizelge 3.5'teki örnek olay kaydının son iki sütunu olan kim tarafından ve hangi zamanda yapıldığı bilgileri önemsiz olduğundan işleme alınmaz.

Çizelge 3.5 Örnek bir olay kaydı

Durum ID	Aktivite ID	İşlemi Yapan Kişi	Yapılma Zamanı
Durum 1	Aktivite A	John	9-3-2004:15.01
Durum 2	Aktivite A	John	9-3-2004:15.12
Durum 3	Aktivite A	Sue	9-3-2004:16.03
Durum 3	Aktivite D	Carol	9-3-2004:16.07
Durum 1	Aktivite B	Mike	9-3-2004:18.25
Durum 1	Aktivite H	John	10-3-2004:9.23
Durum 2	Aktivite C	Mike	10-3-2004:10.34
Durum 4	Aktivite A	Sue	10-3-2004:10.35
Durum 2	Aktivite H	John	10-3-2004:12.34
Durum 3	Aktivite E	Pete	10-3-2004:12.50
Durum 3	Aktivite F	Carol	11-3-2004:10.12
Durum 4	Aktivite D	Pete	11-3-2004:10.14
Durum 3	Aktivite G	Sue	11-3-2004:10.44
Durum 3	Aktivite H	Pete	11-3-2004:11.03
Durum 4	Aktivite F	Sue	11-3-2004:11.18
Durum 4	Aktivite E	Clare	11-3-2004:12.22
Durum 4	Aktivite G	Mike	11-3-2004:14.34
Durum 4	Aktivite H	Clare	11-3-2004:14.38

Her durum için;

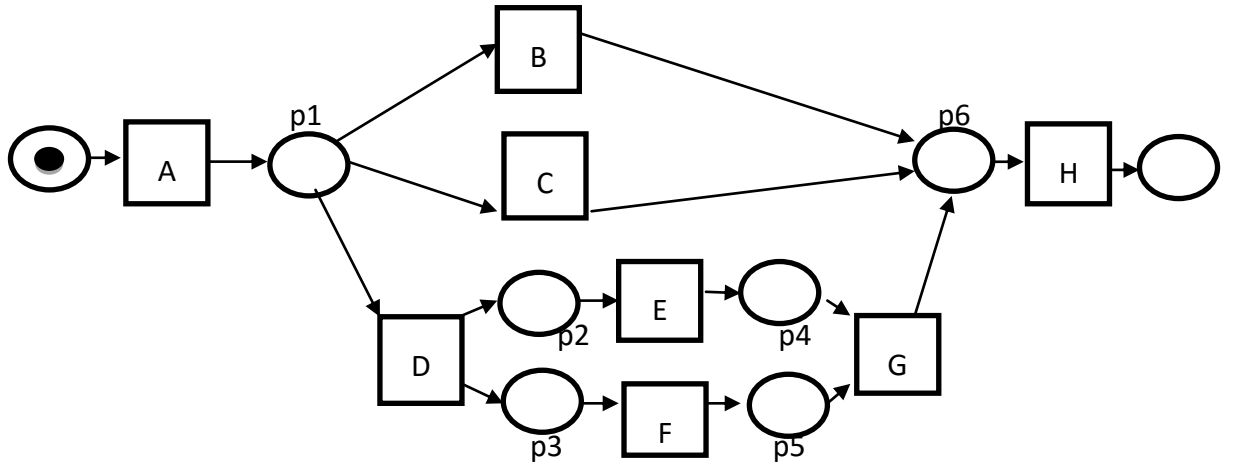
Durum 1 { A,B,H}

Durum 2 {A,C,H}

Durum 3 {A,D,E,F,G,H}

Durum 4 {A,D,F,E,G,H}

olay izleri (event trace) vardır. A aktivitesinden sonra D, B ya da C aktiviteleri gelir. D aktivitesinden sonra ise E ve F ya da F ve E aktiviteleri gelir. Bu durumda E ve F aktiviteleri eş zamanlı olarak düşünülmelidir. E ve F aktiviteleri çalıştıktan sonra G aktivitesi geldiğinden dolayı E ve F aktiviteleri paralel olarak kabul edilir. Bu durumda Çizelge 3.5'deki olay kaydına bakılarak aşağıdaki Şekil 3.17'deki petri ağı oluşur.



Şekil 3.17 Çizelge 3.5 göre oluşan petri ağı

3.3.4.1 Genetik Algoritma için Nedensel Matris (Causal Matrix)

Genetik algoritma ile popülasyondaki bireylere genetik operatörler olan seçim, çaprazlama ve mutasyon yöntemleri uygulanarak uygunluğu yüksek bireyler oluşturur. Sezgisel algoritmada Şekil 3.17'deki petri ağında görülmeyen yerler vardır ve başlangıç popülasyonu oluşturmak oldukça zordur. Ayrıca sezgisel algoritmada seçme, çaprazlama ve mutasyon operatörlerini kullanmak zorlaşır. Bu sebeple sezgisel

algoritmadaki eksikliği gidermek için genetik algorithmada kullanılmak üzere nedensel matris oluşturulmuştur.

Çizelge 3.6 Nedensellik matrisi

GİRİŞ

	Start	A	A	A	D	D	EAf	BVCVG	
→	A	B	C	D	E	F	G	H	ÇIKIŞ
A	0	1	1	1	0	0	0	0	BVCVD
B	0	0	0	0	0	0	0	1	H
C	0	0	0	0	0	0	0	1	H
D	0	0	0	0	1	1	0	0	E
E	0	0	0	0	0	0	1	0	G
F	0	0	0	0	0	0	1	0	G
G	0	0	0	0	0	0	0	1	H
H	0	0	0	0	0	0	0	0	End

Şekil 3.17'deki petri ağına göre Çizelge 3.6'daki nedensel matris oluşturulur [18,19]. Bu petri ağında 8 aktivite olduğundan 8x8 lik yani nxn'lik nedensel matris oluşturulur. Bu matriste satır ve sütunlar aktiviteleri ifade eder ve ilişkiler ikili sayı sistemi ile gösterilir. İki aktivite arasında doğrusal ilişki varsa matrisin ilgili hücresi 1 olurken, doğrusal ilişki yoksa 0 değerini alır. Aktivitelerin giriş (input) değerleri Çizelge 3.6'nın en üst satırında gösterilmiştir. Ayrıca çıkış (output) değerleri ise Çizelge 3.6'nın en sağdaki sütunda verilmiştir [18,19]. Çizelge 3.7'de Çizelge 3.6'nın kodlanmış kısa hali verilmiştir. Ve (And) işlemi $\wedge$ ile gösterilirken, Veya (Or) işlemi ise V ile gösterilir.

Çizelge 3.7 Nedensel matrisin kodlanmış kısa ve öz hali

AKTİVİTE	GİRİŞ	ÇIKIŞ
A	{ }	{ {B, C, D} }
B	{{A}}	{{H}}
C	{{A}}	{{H}}
D	{{A}}	{{E}, {F}}
E	{{D}}	{{G}}
F	{{D}}	{{G}}
G	{{E}, {F}}	{{H}}
H	{ {B, C, G} }	{ }

3.3.4.2 Genetik Algoritmada Ayırıştırma (Parsing) İşlemi

Örnek olarak Çizelge 3.5'teki durum 3 için ayırıştırma işlem adımları Çizelge 3.8'de verilmiştir [18]. Durum 3'ün olay izleri A, D, E, F, G, H aktivitelerinden oluşur. Her adımda bir aktivite ayırıştırılır. Çizelge 3.8'e göre ayırıştırılan aktivite kalın kırmızı renkte gösterilmektedir. Sol sütun ayırıştırılan aktiviteyi gösterirken; sağ sütun ise ilk ayırıştırma işlemi sonucu etkilen aktivitelerin işaretini gösterir. Ayrıca sağ sütundaki aktiviteler dışında başlangıç yerini göstermek için start ve bitiş yerini göstermek için ise end eklenmiştir. Ayırıştırma işlemi sonucu etkilenen aktivite işareti 1 olurken; etkilenmeyen aktivite işaretleri 0 olur.

Çizelge 3.8 Durum 3 için ayrıştırma adımları

Adım	Ayrıştırılan aktivite	Ayrıştırma işlemi sonucu aktivitelerin işareti
1	A ,D,E,F,G,H	A:B=0,C=0,D=0 B:H=0 F:G=0 C:H=0 G:H=0 D:E=0, F=0 start=1 E:G=0 end=0
2	A, D ,E,F,G,H	A:B=1,C=1,D=1 B:H=0 F:G=0 C:H=0 G:H=0 D:E=0, F=0 start=0 E:G=0 end=0
3	A,D, E ,F,G,H	A:B=0,C=0,D=0 B:H=0 F:G=0 C:H=0 G:H=0 D:E=1, F=1 start=0 E:G=0 end=0
4	A,D,E, F ,G,H	A:B=0,C=0,D=0 B:H=0 F:G=0 C:H=0 G:H=0 D:E=0, F=1 start=0 E:G=1 end=0

Çizelge 3.8 Durum 3 için ayrıştırma adımları (devamı)

5	A,D,E,F, G ,H	A:B=0,C=0,D=0 B:H=0 F:G=1 C:H=0 G:H=0 D:E=0, F=0 start=0 E:G=1 end=0
6	A,D,E,F,G, H	A:B=0,C=0,D=0 B:H=0 F:G=0 C:H=0 G:H=1 D:E=0, F=0 start=0 E:G=0 end=0
7	A,D,E,F,G,H	A:B=0,C=0,D=0 B:H=0 F:G=0 C:H=0 G:H=0 D:E=0, F=0 start=0 E:G=0 end=1

Çizelge 3.8 'de gösterilen ayrıştırma adımları aşağıda detaylı olarak ifade edilmiştir.

Adım1: A aktivitesi ayrıştırıldığında giriş ifadesi yani start 1 olur.

Adım2: D aktivitesi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan A'nın bağlı olduğu aktivitelerin çıkış değerleri 1 olur. (B=C=D=1) Aynı zamanda bir önceki ayrıştırılan aktivitenin (A aktivitesinin start ifadesi) güncellenerek 0 değerini alır.

Adım3: E aktivitesi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan D'nin bağlı olduğu aktivitelerin çıkış değerleri 1 olur (E=F=1). Aynı zamanda bir önceki ayrıştırılan D aktivitesinin ayrıştırılan aktiviteleri (B,C,D) güncellenerek 0 değerini alır.

Adım4: F aktivitesi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan E'nin bağlı olduğu aktivitelerin çıkış değerleri 1 olur ($G=1$). Aynı zamanda bir önceki ayrıştırılan E aktivitesinin ayrıştırılan aktiviteleri (E ve F) güncellenerek 0 değerini alması gerekirken VE işleminden dolayı ve F aktivitesi ayrıştırıldığı için $F=1$ olur ve $E=0$ olur.

Adım5: G aktivitesi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan F'nin bağlı olduğu aktivitelerin çıkış değerleri 1 olur ($F:G=1$). Aynı zamanda bir önceki ayrıştırılan F aktivitesinin ayrıştırılan aktiviteleri $E:G=0$ olması gerekirken G çalıştığında dolayı $E:G=1$ olur.

Adım6: H aktivitesi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan G'nin bağlı olduğu aktivitelerin çıkış değerleri 1 olur ($G:H=1$). Aynı zamanda bir önceki ayrıştırılan G aktivitesinin ayrıştırılan aktiviteleri (E ve F) güncellenerek $E:G=0$ ve $F:G=0$ olur.

Adım7: End işlemi ayrıştırıldığında bir önceki ayrıştırılan aktivite olan H'nin bağlı olduğu aktivitelerin çıkış değerleri 1 olur ($end=1$). Aynı zamanda bir önceki ayrıştırılan H aktivitesinin ayrıştırılan aktiviteleri (G) güncellenerek $G:H=1$ olur.

3.3.4.3 Genetik Algoritmada Çaprazlama

Çaprazlama iki farklı birey arasında gen alışverişi yaparak yeni iki yavru bireyin oluşturulmasıdır. Çaprazlama aşağıdaki gibi anlatılmaktadır.

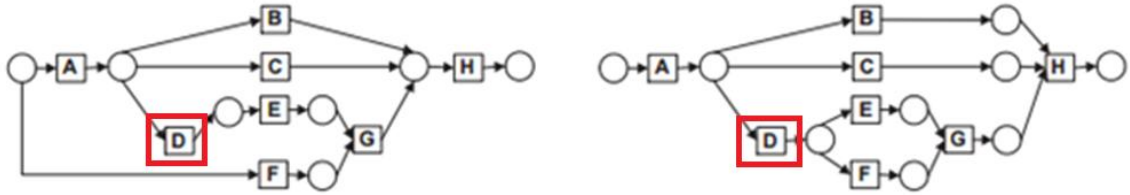
- Çaprazlama noktası her iki bireyde rastgele aktiviteler seçilir.
- Her iki birey için bir değişim noktası (swap) belirlenir.
- Bu değişim noktaları kendi aralarında çaprazlanarak değiştirilir ve yeni yavru bireyler oluşturulur.

Çizelge 3.6'yı göz önünde bulundurularak rastgele iki farklı bireyin oluşturulması Çizelge 3.9 'da gösterilmiştir.

Çizelge 3.9 Rastgele üretilen iki birey

Birey 1			Birey 2		
AKTİVİTE	GİRİŞ	ÇIKIŞ	AKTİVİTE	GİRİŞ	ÇIKIŞ
A	{ }	{ {B, C, D} }	A	{ }	{ {B, C, D} }
B	{{A}}	{{H}}	B	{{A}}	{{H}}
C	{{A}}	{{H}}	C	{{A}}	{{H}}
D	{{A}}	{{E}}	D	{{A}}	{{E, F}}
E	{{D}}	{{G}}	E	{{D}}	{{G}}
F	{ }	{{G}}	F	{{D}}	{{G}}
G	{{E, F}}	{{H}}	G	{{E, F}}	{{H}}
H	{ {C, B, G} }	{ }	H	{ {C}, {B}, {G} }	{ }

Şekil 3.18’de Çizelge 3.9’a göre önceden rastgele üretilen iki bireyin petri ağları verilmiştir.



Şekil 3.18 Rastgele üretilen iki bireyin petri ağı

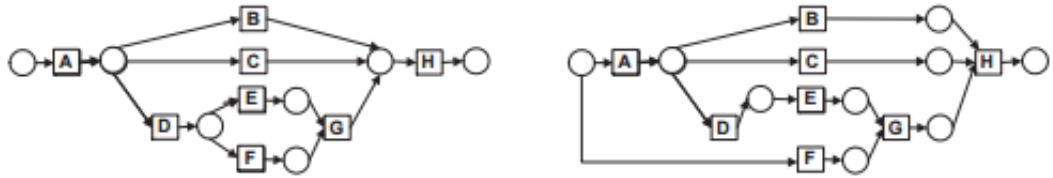
Şekil 3.18’de D aktivitesinin çaprazlama noktasına göre oluşan yavru bireyler Çizelge 3.10’da gösterilmiştir.

Çizelge 3.10 Çaprazlama sonucu oluşan iki yavru birey

Yavru 1		
AKTİVİTE	GİRİŞ	ÇIKIŞ
A	{ }	{ {B, C, D} }
B	{{A}}	{{H}}
C	{{A}}	{{H}}
D	{{A}}	{{E, F}}
E	{{D}}	{{G}}
F	{{D}}	{{G}}
G	{{E}, {F}}	{{H}}
H	{ {C, B, G} }	{ }

Yavru 2		
AKTİVİTE	GİRİŞ	ÇIKIŞ
A	{ }	{ {B, C, D} }
B	{{A}}	{{H}}
C	{{A}}	{{H}}
D	{{A}}	{{E}}
E	{{D}}	{{G}}
F	{ }	{{G}}
G	{{E}, {F}}	{{H}}
H	{ {C}, {B}, {G} }	{ }

Çizelge 3.10'da çaprazlama sonucu oluşan iki yavru bireyin petri ağları Şekil 3.19'de gösterilmiştir.



Şekil 3.19 Çaprazlama işleminin gerçekleştirilmesi

3.4 Süreç Madenciliği Keşif Algortimalarının Karşılaştırma Kriterleri

Algortimaların başarımını ölçmek için birçok farklı kriterler kullanılmaktadır. Bu kriterler uygunluk (fitness), doğruluk (accuracy), hassaslık (precision), basitlik (simplicity), genelleme (generalization) ve zaman ölçütleridir. Tez kapsamında literatürde yaygın kullanılan uygunluk başta olmak üzere doğruluk ve zaman kriterleri süreç madenciliği keşif algortimaları için karşılaştırılmıştır.

Uygunluk fonksiyonu problemin çözümünü sağlayan en uygun bireyin bulunması için hesaplanır. Alfa ve Sezgisel algoritma için uygunluk fonksiyonu **F** ile gösterilmiştir.

Alfa Algoritması için uygunluk fonksiyonu Eşitlik 3.3'deki gibi bulunur.

$$F = \frac{1}{2} * \left(\frac{(s-y)}{s} \right) + \frac{1}{2} * \left(\frac{(s-k)}{s} \right) \quad (3.3)$$

Eşitlik 3.3'deki **s** toplam süreçleri, **y** pars edilemeyen süreçleri, **k** ise pars edilecek kalan süreçleri ifade eder.

Sezgisel Algoritması için uygunluk fonksiyonu Eşitlik 3.4'deki gibi bulunur.

$$F = \frac{3}{5} * \left(\frac{(s-(y+t))}{s} \right) + \frac{2}{5} * \left(\frac{(s-k)}{s} \right) \quad (3.4)$$

Eşitlik 3.4'deki **s** toplam süreçleri, **y** pars edilemeyen süreçleri, **t** tekrar eden süreçleri, **k** ise pars edilecek kalan süreçleri ifade eder.

Genetik Algoritma için uygunluk fonksiyonu F_S ve F_C olmak üzere iki şekilde hesaplanır. Bu hesaplamalar Eşitlik 3.5 ve Eşitlik 3.6'daki gibi hesaplanır. Eşitlik 3.5 ve Eşitlik 3.6'da **PEAS** pars edilen aktivite sayısını, **TAS** tüm aktivitelerin sayısını, **TTİS** tüm tamamlanan izlerin sayısını, **TİS** tüm izlerin sayısını, **TUOTS** tamamlanması uygun olan tüm izlerin sayısını ifade eder.

$$F_S = 0.2x \frac{PEAS}{TAS} + 0.3x \frac{TTİS}{TİS} + 0.5x \frac{TUOTS}{TİS} \quad (3.5)$$

$$F_C = 0.4x \frac{PEAS}{TAS} + 0.6x \frac{TUOTS}{TİS} \quad (3.6)$$

Burada F_C F_S ye göre daha avantajlıdır. Çünkü F_S' 'de tamamlanmayan durumlar olduğu zaman uygunluk fonksiyonu etkili sonuçlar vermez. Ayrıca F_S' 'de ayırıştırma hatası oluştuğunda ayırıştırma durdurulur. Bu nedenle tez çalışmasında genetik algoritma sonuçları için F_C uygunluk fonksiyonu kullanılmıştır.

Doğruluk oranı ise doğru pars edilen süreçlerin sayısının toplam süreçlerin sayısına oranı ile bulunur. Tüm süreç algoritmaları için aynı şekilde hesaplanmıştır.

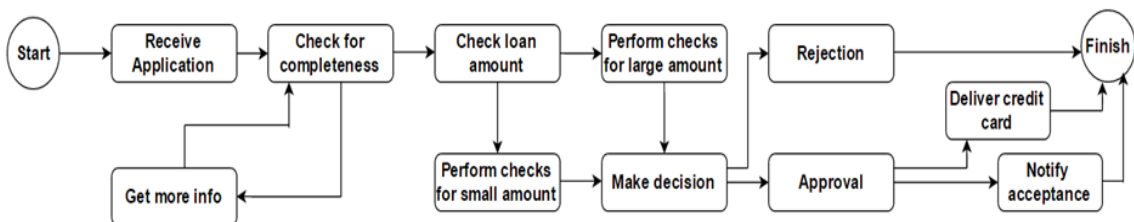
Zaman kriteri ise algoritmaların ortalama çalışma süresini ifade eder.

SÜREÇ MADENCİLİĞİ SİSTEMİNİN GERÇEKLENMESİ

Bu bölümde otomatik veri üretici ile veri üretebilen ve üretilen verilerin olay kayıtlarına süreç madenciliği keşif algoritmaları uygulanarak süreçler hakkında kullanıcılara anlamlı bilgiler sunabilen gerçek hayata uygun olacak şekilde bir süreç madenciliği sistemi tasarlanmıştır. Bu sistemin arayüzü veri üretici ve süreç keşfi olmak üzere iki ana bölümden oluşmaktadır. Veri üretici bölümü süreç modelleme, simülasyonu, analizi, takım karşılaştırılması olmak üzere dört modülden oluşmaktadır. Süreç keşfi bölümü ise alfa, sezgisel, genetik algoritmalar olmak üzere farklı algoritmaları içeren ayrıca süreçlerin animasyon bölümlerinden oluşmaktadır. Bu sistem gerçekleştirirken, MS SQL veritabanı yönetim sistemi ve Visual Studio 2013 ortamı kullanılmıştır.

4.1 Süreç Modeli

Süreç modeli oluştururken herhangi bir süreç modeli örnek olarak incelenebilir. Şekil4.1'de adımları gösterilen kredi kartı başvuru süreci incelenmiştir.



Şekil 4.1 Örnek Kredi kartı başvuru süreci

Oluşturulan sistemde Şekil 4.2'de verildiği gibi başlangıç ve bitiş noktalarına sahip süreçler girilir. Bu çalışmada girilen süreçteki adımlar istenildiği kadar oluşturulabilir ve dallanmalara ayrılmanın yanı sıra, kendi içlerinde çevrimler oluşturabilir. Kullanıcı dallanmalara uğrayan adımlar için yüzdelik dilimler vererek üretilen verilerin kullanıcının istediği oranlara yakın veriler üretmesi sağlanabilmektedir. Kullanıcı sürecin gidiş yolunu belirledikten sonra her adım için bir maliyet ve süre seviyelendirmesi yapar. Aynı zamanda adımları gerçekleştiren kişinin pozisyonlarını da belirleyebilmektedir. Belirlenen pozisyonlar, seçilen adımlara atanarak daha sonradan üretilen verilerde o pozisyondaki kişilerle alakalı analizler yapmak mümkün olacaktır.

Örnek Kredi Kartı Başvuru Süreci

Add Process step

Please enter the steps to be used in the process and press Add button

ID	Step Name
0	Start
1	Receive Application
2	Check for completeness
3	Get more Info
4	Check loan amount
5	Perform checks for small amount
6	Perform checks for large amount
7	Make decision
8	Start Approval
9	Notify Acceptance
10	Deliver Credit Card
11	Complete Approval
12	Notify Rejection
13	Finish

Check for completeness

Input

Output

Input Steps

Output steps

Start

Check for completeness

Duration

Cost

Short

Low

Role

Person Count

employee

Select

Details of Ways

Create Process

Şekil 4.2 Süreç oluşturma ekranı

Bu adımın kullanıcıya kolaylık sağlaması ve birbirinden farklı fakat birbirine benzer süreçler oluşturup aralarındaki farkları görebilmesi açısından veri üretme aşamasına geçmeden önce mevcut süreçleri kullanabilmesini sağlayan bir bölüm de mevcuttur. Kullanıcı bir süreç oluşturduğu zaman sistemdeki mevcut süreçler sayfasına o süreç eklemiş olup, veri üretmeye hazır hale gelir.

4.2 Simülasyon

Simülasyon modülü olay günlüklerinin oluşturulduğu bölümdür. Olay günlükleri oluşturulurken kullanıcının seçtiği süreç modeline yönelik veriler üretilir. Simülasyon ekranından kullanıcıdan bazı parametrelerin tanımlanması istenir. Bu parametreler arasında sürecin başlama tarihi ve mesai başlama saati ile mesai bitiş saati sürecin gün içerisinde uygulanacağı alan bulunmaktadır. Oluşturulan sistem, kullanıcının istediği şekilde model oluşturmaya olanak verecek şekilde esnek bir yapıda olup istenildiği kadar süreç tamamlanma sayısı (Track Count) ve takım sayısı (Team Count) tanımlanabilir. Takım sayısı girildiği zaman bu takımlara ait üretilen verilerde her takımın aynı tarih ve saatte işleme başladığı varsayılarak paralel olarak süreçle ilgili olay kayıtları üretilir. Kullanıcının belirli bir süreçte birbirinden farklı süreç tamamlanma sayısı ve takım sayısında veri üretme ihtiyacı olması kaçınılmaz olduğundan dolayı her ürettiği simülasyona bir isim vermesi istenmektedir. Bu sayede kullanıcıya seçtiği süreç için farklı takım sayıları için farklı analizler yaparak, optimum takım sayısına karar verebilme olanağı verilmiştir.

Simulation Details			
Teams	Tracks	Process Steps	Steps Details
1	185 186 187 188 189 190 191 192 193 194 195 196	Receive Application Check for completeness Check loan amount Perform checks for amount Make decision Start Approval Notify Acceptance Complete approval Finish	Steps Name: Make decision Resource: Doğan Durak Role: Manager Cost: 7,176,148 \$ Start Date: 4.6.2015 16:57:00 Duration: 9 Min

Şekil 4.3 Simülasyon sonucu üretilen bilgiler

Simülasyon sonrasında üretilen verilerin incelenmesi için Şekil 4.3 gibi bir arayüz tasarlanmıştır. Kullanıcı üretilen verilerde hangi takımın hangi Track de hangi adımlardan gidildiğini gözlemleyebilir. Seçilen adımın başlangıç zamanı, tamamlanma süresi ve hangi çalışanın gerçekleştirdiği bilgilerinin yanı sıra maliyet bilgileri gibi adımlar hakkında detaylı bilgilere ulaşmak mümkündür. Kullanıcı ürettiği verilerin olay kayıtlarını XML formatında çıktısını alıp, sistemin keşif modülündeki mevcut süreç

madenciliği keşif algoritmaları ile süreçlerin uygunluğu, doğruluğu ve gerçekleşme süreleri hakkında anlamlı bilgilere ulaşılabilir.

4.3 Analiz

Analiz modülünde, simülasyon modülünde oluşturulan olay kayıtlarından anlamlı bilgiler çıkartmak amaçlanmıştır. Bu ekranda daha önceden yapılmış olan süreç simülasyonları listelenir ve kullanıcı analizi yapılmak istenen süreci seçer. Seçilen model sonrasında kullanıcıya Şekil 4.4'de gösterildiği gibi özet bilgiler verilmiştir. Sol tablo süreç modeline ait genel bilgileri içerirken sağ tablo ise süreç modelinin seçilen takımına ait bilgileri gösterir. Oluşturulan özet analizde seçilen takıma ait veriler, sürecin sahip olduğu ortalama veriler ile karşılaştırılarak takımlar arasında genel olarak bir performans karşılaştırması yapılabilmektedir.

Step Name	Max Cost	Mean Cost	Min Cost	Max Duration	Mean Duration	Min Duration
Receive Application	0,94	0,52	0,11	0	0	0
Check for completeness	176,94	45,65	0,04	67	18	1
Get more info	17,96	6,84	0	22	7	1
Check loan amount	17,76	7,83	0,38	21	5	1
Perform checks for amount	95,17	53,47	10,84	67	31	11
Perform checks for small amoun	173,57	90,99	12,78	68	30	13
Make decision	16,98	5,62	0,01	21	7	1

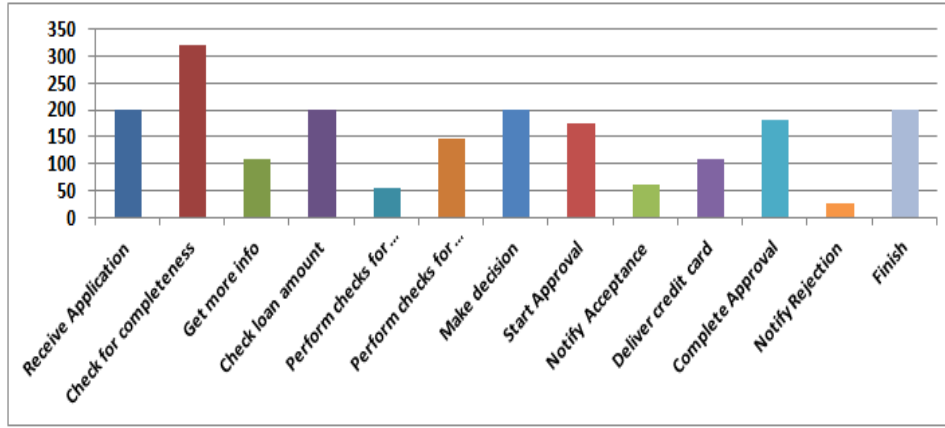
Step Name	Max Cost	Mean Cost	Min Cost	Max Duration	Mean Duration	Min Duration
Receive Application	0,93	0,62	0,12	0	0	0
Check for completeness	175,32	38,76	0,4	67	17	5
Get more info	15,9	6,51	0	22	8	2
Check loan amount	16,88	7,84	0,38	20	6	1
Perform checks for amount	89,46	59,85	26,41	67	30	11
Perform checks for small amoun	173,57	78,28	16,81	30	21	17
Make decision	16,98	8,18	3,64	20	6	2

Şekil 4.4 Analiz sonucu çıkarılan özet bilgiler (sol genel ve sağ takım)

4.4 Takım Karşılaştırması

Adımlara ait Şekil 4.5'de olduğu gibi genel, takım bazlı ve seçilen *Track'lere* ait süre ve maliyet dağılımlarını gösteren grafikler üretilmektedir. Üretilen bu grafikler sayesinde süreç içerisindeki en fazla maliyet ve süreye sahip adımlar belirlenerek sürecin geliştirilmesi açısından yapılacak değişiklikler saptanabilmektedir.

Bununla birlikte takımlar arasındaki *Track'ler* için en kısa sürede ve en uzun sürede bitiren takımları saptamak, hangi takımın hangi performans ile çalıştığını görmek mümkündür. Hangi *Track'de* hangi takımın o süreci ne kadar bir sürede bitirdiğine ait bilgilere ulaşmak mümkündür.



Şekil 4.5 Adım Frekans analiz verileri

Takım karşılaştırma yapısı sayesinde çok kanallı (multi-thread) mimariye uygun üretilen verilerden anlamlı analizler yapılabilmektedir. Sistemin esnekliğinin bir sonucu olarak istenilen simülasyona ait farklı zamanlarda ve farklı takım sayıları için veriler üretilerek karşılaştırma yapılabilir.

BÖLÜM 5

DENEYSEL SONUÇLAR

Bu tez çalışmasında geliştirilen veri üretici simülasyonun ve süreç madenciliği keşif algoritma sonuçları aşağıdaki bölümlerde detaylı bir şekilde incelenmiştir.

5.1 Veri Üretici Simülasyon Sonuçları

Hasta Takip Süreç adımları ile ilgili bir model oluşturma ekranı Şekil 5.1'de gösterilmiştir.

Process Mining Discover Play Model Simulation Analyze Team Comparison About Exit

Creating a New Process Model

Hasta Takip Süreci

Add Process step Please enter the steps to be used in the process and press

ID	Step Name
0	Başla
1	Hasta bilgilerini al
2	Muayene Et
3	Laboratuvar örneği al
4	Sonuç
5	İyi
6	Kötü

Input Output

Input Steps Output Steps

Duration Cost

Select Duration Select Cost

Role Person Count

Select

Details of Ways Create Process

Şekil 5.1 Örnek Hasta Takip Süreç modeli oluşturma

Oluşturulan örnek Hasta Takip Süreç modeline ait genel bilgiler Şekil 5.2'de verilmiştir.

Hasta Takip Süreci

Add Process step

Please enter the steps to be used in the process and press the button

ID	Step Name
0	Başla
1	Hasta bilgilerini al
2	Muayene Et
3	Laboratuvar örneği al
4	Sonuç
5	İyi
6	Kötü
7	Hasta Yatışı
8	Bitiş

Finish

Clear

Hasta Yatışı

Input

Output

Input Steps

Output Steps

Laboratuvar örneği al

İyi
Kötü

Duration

Cost

Medium

Medium

Role

Person Count

Doktor

Select

Details of Ways

Create Process

Şekil 5.2 Oluşturulan Hasta Takip Süreç modelinin genel özeti

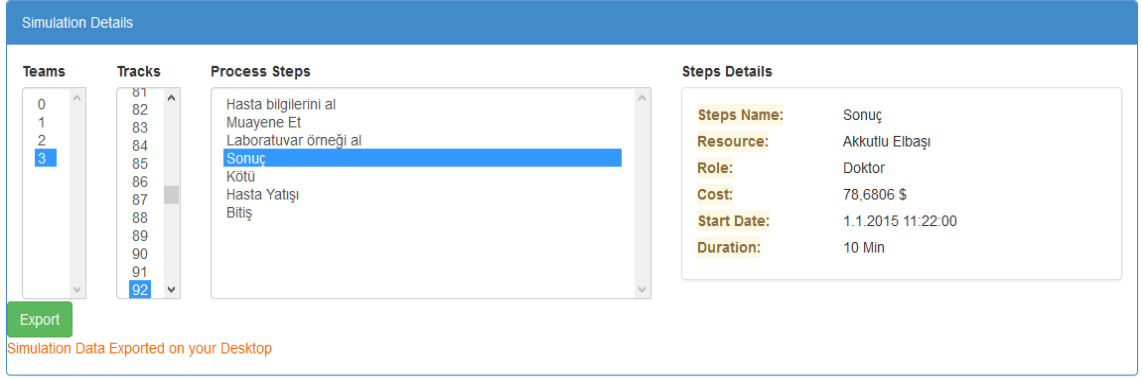
Hasta Takip Süreç modelinin seçilen tarih ve saatte göre 4 takımlı ve 170 track'e sahip simülasyon ekranı Şekil 5.3'te görüldüğü gibi oluşturulmuştur.

Simulation Screen for Hasta Takip Süreci

4	Start	170
12/10/2014 09:00	Hasta Takip Süreci Simülas	17:00

Şekil 5.3 Oluşturulan Hasta Takip Süreç modelinin simülasyonunu oluşturma

Hasta Takip Süreç modeline ait oluşturulan simülasyonda Şekil 5.4'te verilerle ilgili rol, maliyet ve zaman ile ilgili bilgilere ulaşılır. Ayrıca export diyerek simülasyon verileri xml olarak masaüstüne kaydebilebilir.



Şekil 5.4 Oluşturulan Hasta Takip Süreç modelinin simülasyon özeti

Hasta Takip Süreç modeline ait genel(sol) ve takım(sağ) analizi Şekil 5.5'de verilmiştir.

Summary statistics Hasta Takip Süreci

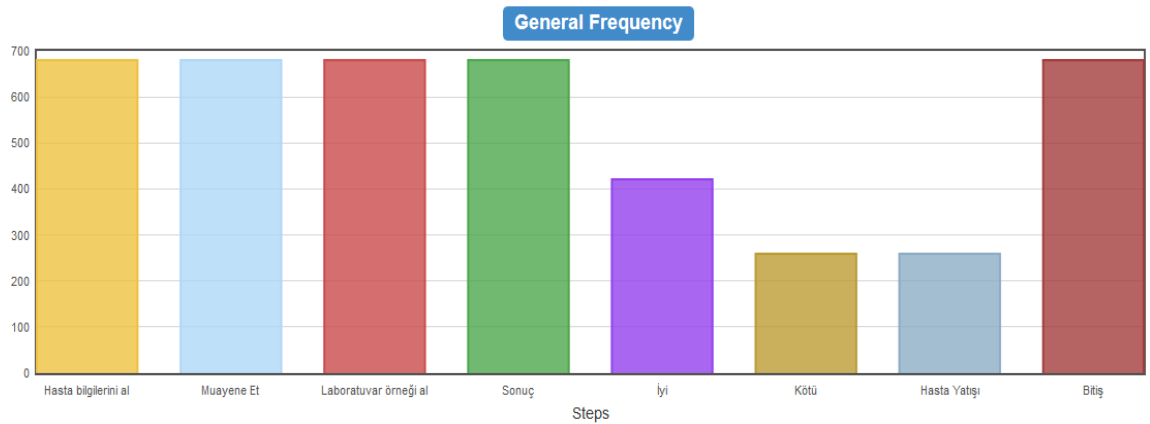
Hasta Takip Süreci S 3 44

Step Name	Max Cost	Mean Cost	Min Cost	Max Duration	Mean Duration	Min Duration
Hasta bilgilerini al	1	0,48	0	0	0	0
Muayene Et	17,99	6,73	0	22	7	1
Laboratuvar örneği al	179,79	65,43	10,48	68	24	10
Sonuç	179,36	65,45	10,04	68	24	10
İyi	179,84	70,55	10,82	68	26	10
Kötü	179,19	67,88	10,84	68	23	10
Hasta Yatışı	17,95	6,39	0	22	6	1
Bitiş	17,93	6,87	0,03	22	7	1

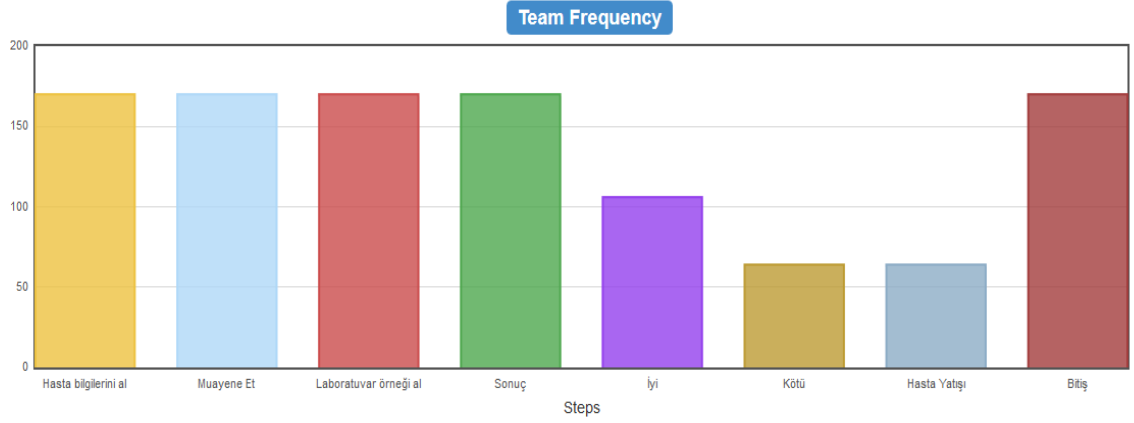
Step Name	Max Cost	Mean Cost	Min Cost	Max Duration	Mean Duration	Min Duration
Hasta bilgilerini al	1	0,5	0,01	0	0	0
Muayene Et	17,99	7,06	0,03	22	8	1
Laboratuvar örneği al	179,69	65,95	10,51	68	24	10
Sonuç	177,59	66,05	10,04	68	23	10
İyi	178,85	79,41	11,34	68	27	10
Kötü	173,69	67,27	12,05	68	21	10
Hasta Yatışı	17,95	6,66	0,11	21	6	1
Bitiş	17,83	7,2	0,11	22	7	1

Şekil 5.5 Hasta Takip Süreç modeline ait analiz sonuçları

Hasta Takip Süreç modeline ait genel frekans grafiği Şekil 5.6'da ve 3.takıma ait frekans grafiği Şekil 5.7'de verilmiştir.

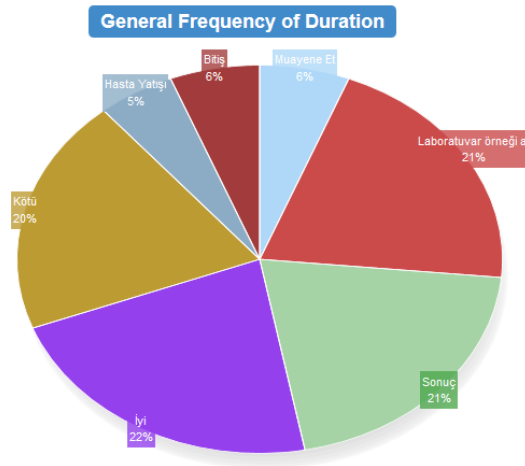


Şekil 5.6 Hasta Takip Süreç modeline ait genel frekans bilgileri

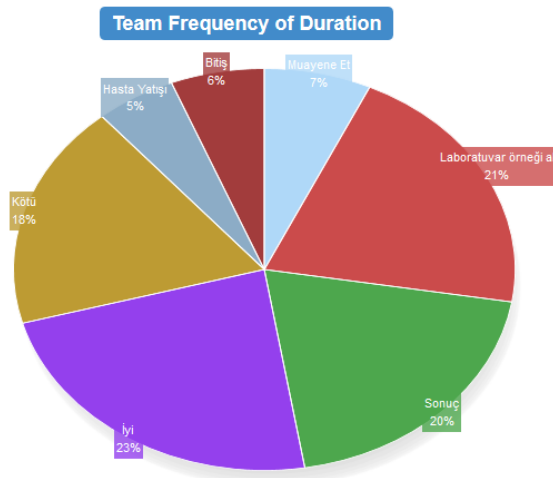


Şekil 5.7 Hasta Takip Süreç modelinin 3.takıma ait frekans bilgileri

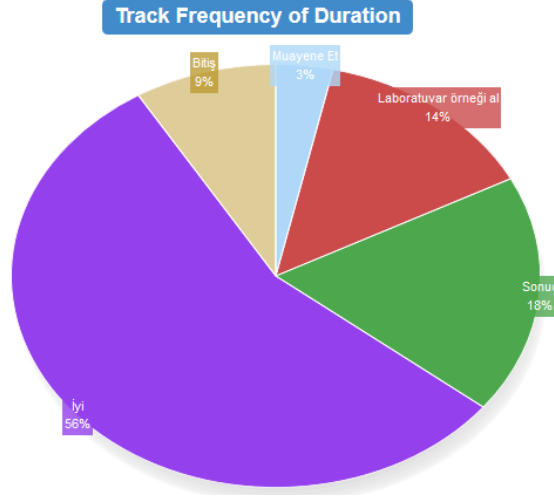
Hasta Takip Süreç modeline ait genel zaman frekans grafiği Şekil 5.8'de ve 3.takıma ait zaman frekans grafiği Şekil 5.9'da ayrıca 3. takımın 44. track'ine ait zaman frekans grafiği Şekil 5.10'da verilmiştir.



Şekil 5.8 Hasta Takip Süreç modeline ait genel zaman frekans grafiği

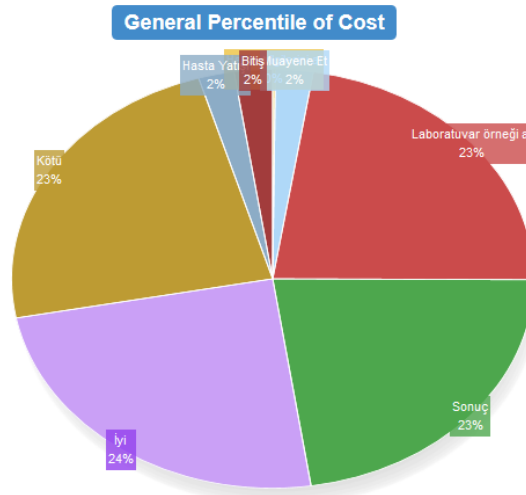


Şekil 5.9 Hasta Takip Süreç modelinin 3.takımına ait zaman frekans grafiği

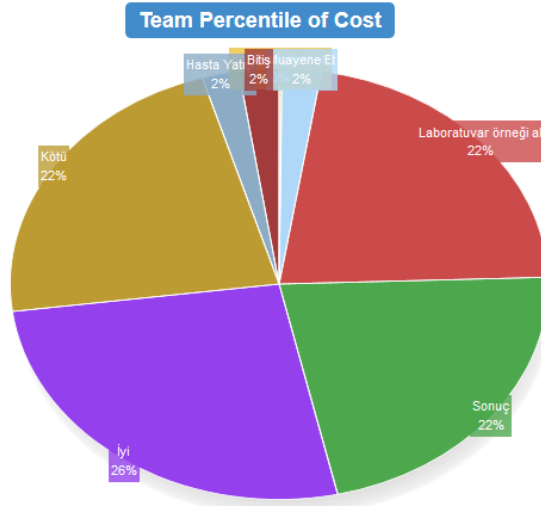


Şekil 5.10 Hasta Takip Sürecinin 3.takımının 44. track'e ait zaman frekans grafiği

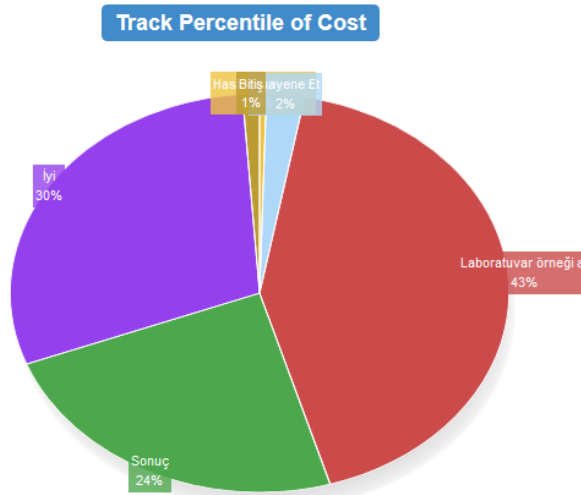
Hasta Takip Süreç modeline ait genel maliyet grafiği Şekil 5.11'de ve 3.takıma ait maliyet grafiği Şekil 5.12'de ayrıca 3. takımın 44. track'ine ait maliyet grafiği Şekil 5.13'de verilmiştir.



Şekil 5.11 Hasta Takip Süreç modeline ait genel maliyet grafiği

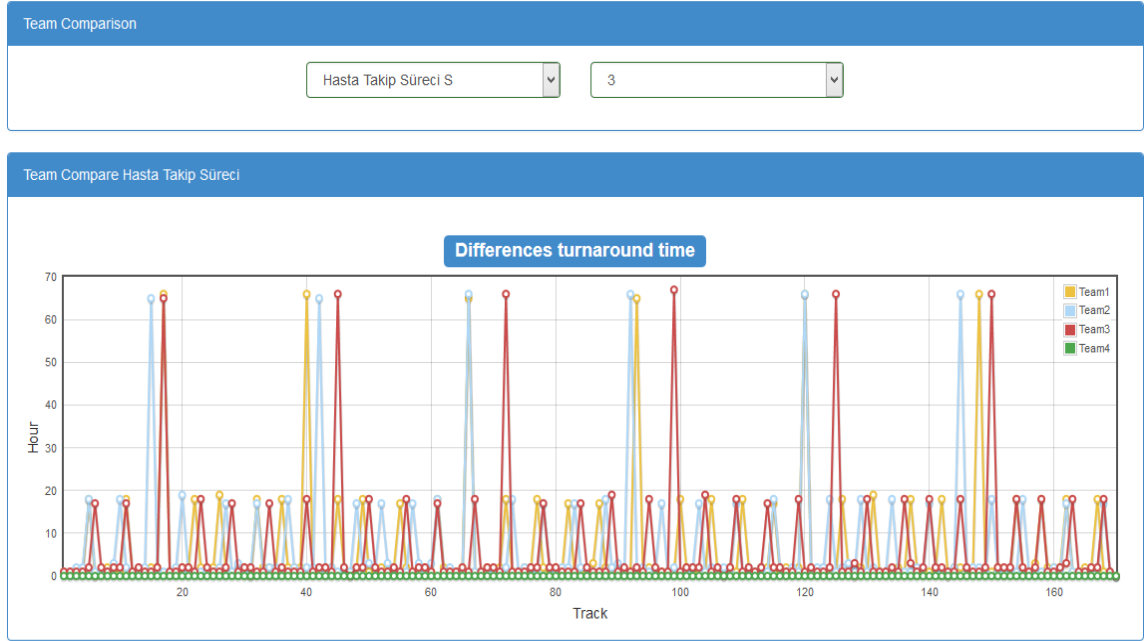


Şekil 5.12 Hasta Takip Süreç modelinin 3.takımına ait maliyet grafiği



Şekil 5.13 Hasta Takip Süreç modelinin 3.takımının 44. track'e ait maliyet grafiği

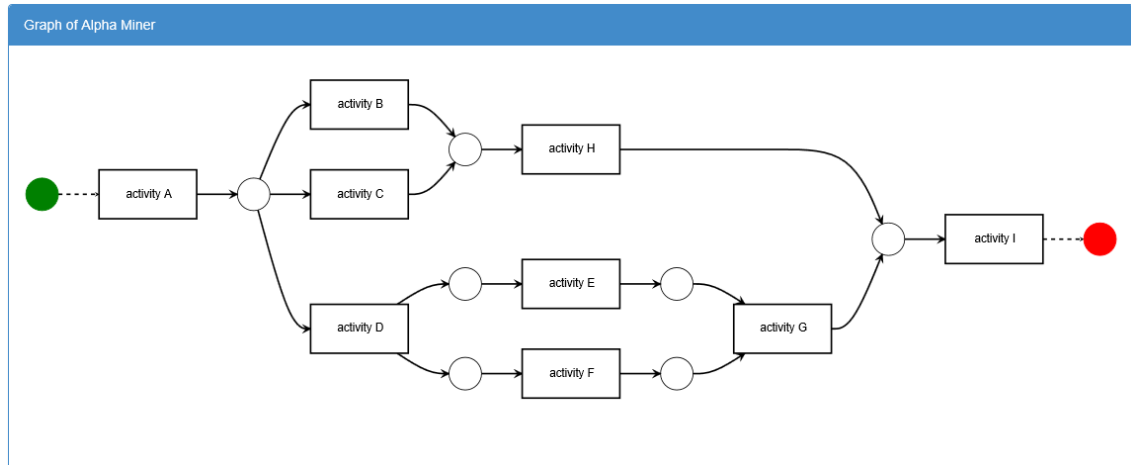
Şekil 5.14'de Hasta Takip Süreç modeline ait takımların kendi aralarında karşılaştırılması verilmiştir.



Şekil 5.14 Hasta Takip Sürec modeline ait takımlar arası karşılaştırma grafiği

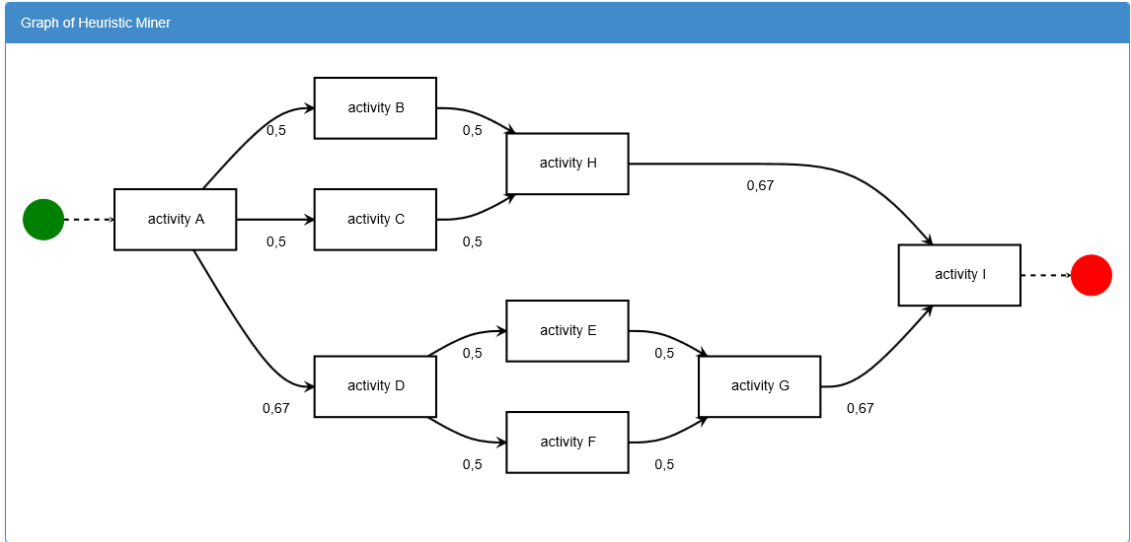
5.2 Süreç Madenciliği Algoritmalarıyla Süreç Keşfi

Simülasyon ile oluşturulan 9 aktivite, 327 durum ve 1255 olay kaydından oluşan örnek kredi veri seti için alfa algoritması ile oluşturulmuş petri ağı Şekil 5.15'de gösterilmiştir.



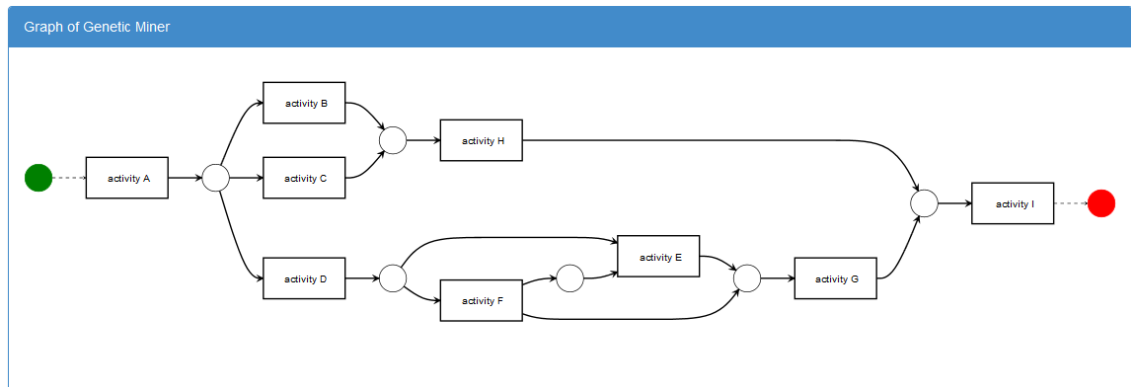
Şekil 5.15 Örnek kredi veri setinin alfa algoritmasıyla oluşturulmuş petri ağı

Örnek kredi veri seti için sezgisel algoritma ile oluşturulmuş petri ağı Şekil 5.16'da gösterilmiştir.



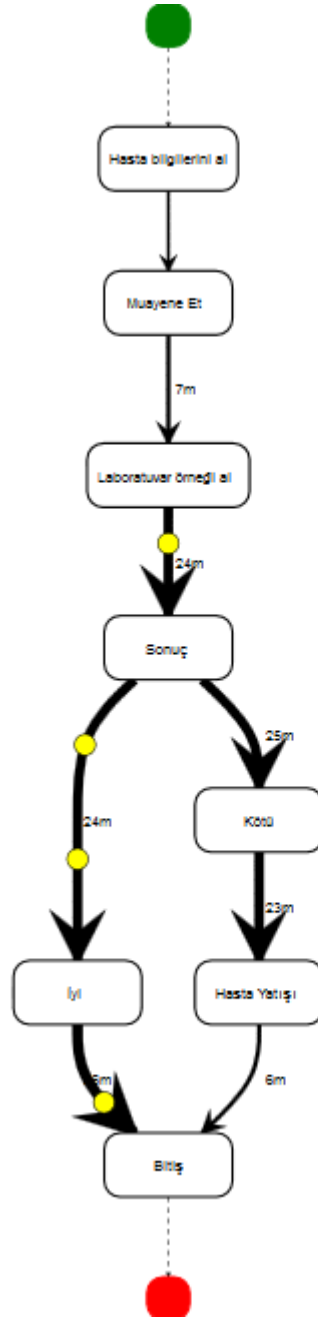
Şekil 5.16 Örnek kredi veri setinin sezgisel algoritma ile oluşturulmuş petri ağı

Örnek kredi veri seti için genetik algoritma ile oluşturulmuş petri ağı Şekil 5.17'de gösterilmiştir.



Şekil 5.17 Örnek kredi veri setinin genetik algoritma ile oluşturulmuş petri ağı

Geliştirilen sistemde süreçlerin akışını görebilmek için bir görselleştirme aracı geliştirilmiştir. Hasta Takip Süreci veri modelinin görselleştirilmesi Şekil 5.18'de gösterilmiştir. Şekildeki kalın çizgiler tıkanıklığı yani sürecin gerçekleşme süresinin uzun olduğunu ince çizgiler ise sürecin gerçekleşme süresinin kısa olduğunu ifade eder.



Şekil 5.18 Hasta Takip Sürecinin görselleştirilmesi

5.3 Süreç Madenciliği Keşif Algoritmalarının Performans Analizi

Çizelge 5.1'de simülasyon kullanılarak üretilen çeşitli sentetik veri setleri üzerinde süreç madenciliği algoritmalarının sonuçları verilmiştir. Sentetik Veri5 (Hasta Takip Verisi) üzerinde bazı aktiviteler silinerek Eksik Sentetik Veri6 (Hasta Takip Verisi) oluşturulmuştur. Bu iki verinin uygunluk değerleri arasındaki değişim Çizelge 5.1 gösterildiği gibi alfa algoritması için uygunluk değeri 0.16 (0.78-0.62) oranında

düşerken, sezgisel algoritmanın uygunluk değeri 0.15 (0.83-0.68) genetik algoritmanın uygunluk değeri ise 0.04 (0.87-0.83) oranında düşmüştür. Eksik veride genetik algoritmanın uygunluk değerindeki değişim diğer algoritmalarla göre daha az düşüş göstermiştir.

Çizelge 5.1 Süreç madenciliği algoritmalarının simülasyon veri setleri üzerindeki sonuçları

Simülasyon Veri Setleri [31]	Uygunluk Değeri			Doğruluk Oranı (%)		
	Alfa	Sezgisel	Genetik	Alfa	Sezgisel	Genetik
Sentetik Veri1(Günlük Aktivite)	1	1	1	100	100	100
Sentetik Veri2(İş başvurusu)	0.97	1	0.99	95.5	97.5	96
Sentetik Veri3(Yüksek Lisans Başvuru)	0.92	0.94	0.97	93	95	97.6
Sentetik Veri4(Kredi Başvurusu)	0.91	0.93	0.96	87.4	89.5	92.5
Sentetik Veri5(Hasta Takip Verisi)	0.78	0.83	0.87	79	84.5	86.7
Eksik Sentetik Veri6(Hasta Takip Verisi)	0.62	0.68	0.83	60	65.3	72.5
Ortalama	0.86	0.89	0.93	85.8	88.6	90.8

Çizelge 5.2'de mevcut [32] çeşitli sentetik veri setleri üzerinde süreç madenciliği algoritmalarının sonuçları verilmiştir. Algoritmaların ortalama uygunluk ve doğruluk değerlerine bakıldığında genetik algoritma sezgisel ve alfa algoritmasına göre başarılı

sonular vermiřtir. Ayrıca sezgisel algoritmanın alfa algoritmasına gre daha iyi sonular verdiėi grlmektedir.

izelge 5.2 Sre madenciliėi algoritmalarının mevcut sentetik veri setleri zerindeki sonuları

Mevcut Sentetik Veri Setleri [32]	Uygunluk Deėeri			Doėruluk Oranı (%)		
	Alfa	Sezgisel	Genetik	Alfa	Sezgisel	Genetik
Sentetik Veri1(Exercis e1)	0.97	1	1	97.5	100	100
Sentetik Veri2(Exercis e2)	0.92	0.98	0.95	95.5	97.5	95
Sentetik Veri3(Exercis e3)	0.87	0.73	0.81	80	70.5	77.5
Sentetik Veri4(Exercis e4)	0.72	0.81	0.84	70	85	86
Sentetik Veri5(Exercis e5)	0.67	0.73	0.81	70	80.5	85.5
Eksik Sentetik Veri6(Exercis e1)	0.52	0.69	0.74	66.4	73.5	78
Ortalama	0.77	0.82	0.85	79.9	84.5	87

izelge 5.3'de gerek veri setleri zerinde sre madenciliėi algoritmalarının sonuları verilmiřtir [33,34]. izelge 5.3'te Eksik Gerek Veri3 (Hasta Bilgisi) olay kaydından rastgele bazı olaylar silinmiřtir. Eksik ve Grltl Gerek Veri5 (Trafik) olay kaydına rastgele bazı olaylar eklenmiř ve silinmiřtir. Bu veri setleri iin alfa algoritmasının bařarımı sezgisel ve genetik algoritmaya gre daha dřktr. Genetik algoritma ise sezgisel algoritmaya gre daha iyi sonular vermiřtir.

Çizelge 5.3 Süreç madenciliği algoritmalarının gerçek veri setleri üzerindeki sonuçları

Gerçek Veri Setleri [33,34]	Uygunluk Değeri			Doğruluk Oranı (%)		
	Alfa	Sezgisel	Genetik	Alfa	Sezgisel	Genetik
Gerçek Veri1 (Teleclaims)	0.69	0.98	0.99	71	86	91
Gerçek Veri2 (Hasta Bilgisi)	0.67	0.84	0.89	68	74	83
Eksik Gerçek Veri3 (Hasta Bilgisi)	0.61	0.78	0.82	0	40	52
Gerçek Veri4 (Trafik)	0.57	0.73	0.62	0	55	64
Eksik ve Gürültülü Gerçek Veri5 (Trafik)	-	-	0.54	-	-	56

BÖLÜM 6

SONUÇ

Süreç madenciliği, süreçlerin keşfedilmesi, analiz edilmesi, modellenmesi ve iyileştirilmesi çalışmalarını kapsar ve bu şekilde süreç verilerinin etkin ve verimli olarak kullanılmasını sağlar. Rekabetçi iş ortamında büyük işletmelerin süreç madenciliği tekniklerini kullanması zorunlu hale gelmiştir.

Günümüz bilgi sistemlerinde iş süreçleri üzerinde oluşan olaylar hakkındaki bilgiler kaydedilir. Bu olay günlükleri incelenerek zaman, maliyet ve süreç hakkında anlamlı bilgiler çıkarılabilmektedir. Çıkarılan bu bilgiler sayesinde süreçleri geliştirmek veya sürecin performansını ölçmek mümkündür.

Bu tez kapsamının ilk aşamasında web tabanlı bir veri üretici geliştirilmiştir. Dört ana modüle sahip bu veri üreticinde; isteğe göre süreç modelleri oluşturulabilir, süreçlere uygun veriler üretilebilir, üretilen veriler ile ilgili analiz ve takım karşılaştırmaları yapılabilir. İlk modülde örnek bir veri modeli oluşturulur. İkinci modül olan simülasyon modülünde ise daha önceden girilen modele uygun olay kayıtları üretilmektedir. Analiz modülünde üretilmiş olan olay kayıtlarından süreçlerle ilgili anlamlı bilgiler çıkartılmaktadır. Son modül olan takım karşılaştırmasında, sürecin paralel olarak takımlara bölünmesi ve takımlar arasındaki performans dağılımları karşılaştırılarak anlamlı bilgiler çıkartılabilir.

Bu tezin ikinci aşamasında mevcut olay kayıtları veya veri üretici ile oluşturulan olay kayıtlarından süreç keşfini sağlayan bir sistem oluşturulmuştur. Veriler MXML ve XES formatında alınıp sisteme yüklendikten sonra alfa, sezgisel ve genetik gibi çeşitli süreç

madenciliği algoritmaları uygulanarak sonuçlar uygunluk, doğruluk ve zaman kriterlerine göre karşılaştırılmıştır.

Alfa ve sezgisel algoritma sonuçları karşılaştırıldığında sezgisel algoritma alfa algoritmasına göre uygunluk ve doğruluk oranları açısından daha başarılı sonuçlar vermiştir. Genetik algortmada ise sezgisel algoritmaya göre genetik işlemler (seçme, çaprazlama ve mutasyon) uygulandığından tamamlanmayan süreçlerle ilgili daha iyi sonuçlar verir.

Sonuçlar zaman kriteri açısından karşılaştırıldığında ise basit veriler için alfa algoritması sezgisel ve genetik algoritmaya göre daha kısa sürede sonuçlanırken, karmaşık veriler için genetik algoritma alfa ve sezgisel algoritmaya göre daha kısa sürelerde sonuçlanır.

Gelecekte bu çalışmaya farklı algoritmalar entegre edilerek sistemin geliştirilmesi hedeflenmektedir. Ayrıca gürültülü veriler, tamamlanmış süreçler gibi süreç madenciliği problemleriyle başa çıkmak için çalışmaların geliştirilmesi öngörülmektedir.

KAYNAKLAR

- [1] Rubin, V., Günther, C. ve Van der Aalst, W.M.P., (2007). "Process Mining Framework for Software Processes", *Software Process Dynamics and Agility*, 169-181.
- [2] Ronny, M. ve Helen, S., (2008). *Process Mining Techniques: An Application to Stroke Care*, IOS Press.
- [3] Stijn, G. ve Jochen D.W., (2011). "Process discovery in event logs: An application in the telecom industry", *Applied Soft Computing* 11, 1697–1710 .
- [4] Weerapong, S., Porouhan P. ve Premchaiswadi, W., (2012). "Process Mining Using σ -Algorithm as a Tool (A Case Study of Student Registration)", 2012 Tenth International Conference on ICT and Knowledge Engineering, 213-220.
- [5] Weijters, A.J.M.M. , Van der Aalst, W.M.P. ve Alves, A.K., (2006). "Process Mining with the Heuristics Miner Algorithm", *BETA Working Paper Series*, WP 166, Eindhoven University of Technology, Eindhoven.
- [6] Saravanan, M.S ve Rama Sree , R.J, (2011). "Evaluation Of Process Models Using Heuristic Miner And Disjunctive Workflow Schema Algorithm For Dyeing Process", *International Journal of Information Technology Convergence and Services (IJITCS)*, 1(3):47-68.
- [7] Koonce, D.A. ve Tsai, S.C. , (2000). "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule", *Pengamon Computers & Industrial Engineering*, 361-374.
- [8] Partington, A., Wynn, M. ve Suriadi, S., (2015). "Process Mining for Clinical Process: A Comparative Analysis of four Australian Hospitals", *ACM Transactions on Management Information System*, 5(4):1-18.
- [9] Ying, L. ve Hui, Z., (2012). "Workflow simulation for operational decision support using event graph through process mining", *Decision Support Systems*, 52(3): 685-697.
- [10] Leoni, M., Van der Aalst, W.M.P. ve Hofstede, A.H.M., (2008). "Visual Support for Work Assignment in Process-Aware Information Systems", *Springer Verlag*, 67-83.

- [11] Agrawal, R., Gunopulos, D. ve Leymann, F. , (1998). "Mining process models from workflow logs" Sixth International Conference on Extending Database Technology, 469–483.
- [12] Ingvaldsen, J. ve Gulla, J., (2012). "Industrial application of semantic process mining", Enterprise Information Systems, 6(2):139-163.
- [13] Nick, G. ve Werner, M., (2012). "Process Mining", WISU - die Zeitschrift für den Wirtschaftsstudenten 7/13, 1-16.
- [14] Van der Aalst, W., Adriansyah, A. ve Medeiros, A.K.A., (2011). "IEEE Task Force on Process Mining, Process Mining Manifesto", Lecture Notes in Business (Information Processing), 99(169-194).
- [15] Şahin, A., "Süreç Madenciliği Manifestosu", <http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:pmm-turkish-v2.pdf>, 10 Ocak 2014.
- [16] Aalst, W.M.P., Dorgen, B.F. ve Mans, R.S., (2007). "ProM 4.0: Comprehensive Support for Real Process Analysis", ICATPN Springer Verlag, 484-494.
- [17] Aalst, W.M.P., (2011). "Process Mining", Discovery, Conformance Enhancement of Business Processes, Springer.
- [18] Medeiros, A.K., Weijters, A.J.M.M. ve Aalst, W.M.P. , (2004). "Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results", BETA Working Paper Series, WP 124, Eindhoven University of Technology, Eindhoven.
- [19] Aalst, W.M.P. , Alves de Medeiros, A.K. ve Weijters, A.J.M.M. , (2005). "Genetic Process Mining", , in: G. Ciardo, P. Darondeau (Eds.), 26th International Conference on Applications and Theory of Petri Nets (ICATPN 2005), Lecture Notes in Computer Science, 3536:48–69.
- [20] Medeiros, A.K.A., Dongen, B.F. ve Aalst, W.M.P., (2004). "Process Mining: Extending the a-Algorithm to Mine Short Loops", BETA Working Paper Series, WP 113, Eindhoven University of Technology, Eindhoven.
- [21] Aalst W.M.P., Weijters, A.J. ve Maruster, L., (2004). "Workflow Mining: Discovering process models from events logs", Knowledge and Engineering IEEE Transactions, 16(9):1128-1142.
- [22] Weijters, A.J.M.M. ve JRibeiro, T.S., (2011). "Flexible Heuristics Miner (FHM)", Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, 310-317.
- [23] Masao, M. ve Tseng, C.C., (1997). "A genetic algorithm for multi-mode resource constrained project scheduling problem", European Journal of Operational Research, 100(1):134-141.
- [24] Berry, M.A ve Linoff, G.L, (1997). "Data Mining Techniques for marketing, Sales and Customer Support, 335-336.

- [25] Koza, J.R., (1995). "Two ways of discovering the size and shape of a computer program to solve a problem", Proceedings of the Sixth International Conference on Genetic Algorithm, 287-294.
- [26] Goldberg, D.E., (1989). "Genetic Algorithms in Search Optimization and Machine Learning", Addison Wesley Publishing Company, Usa.
- [27] Emel, G.G. ve Taskın, Ç., (2002). "Genetik Algoritmalar ve Uygulama Alanları", Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, 21, 1, 129–152.
- [28] Verma, V.K ve Kumar, B., (2014). "Genetic algorithm: an overview and its application", International Journal of advanced studies in Computer Science and Engineering IJASCSE,3(2):21-27.
- [29] Hermawanto, D., (2013). "Genetic Algorithm for Solving Simple Mathematical Equality Problem" Indonesian Institute of Sciences (LIPI), Indonesia.
- [30] Sastry, K.,Goldberg, D. ve Kendall, G., (2005). "Genetic algorithms", Search methodologies: introductory tutorials in optimisation and decision support techniques Berlin: Springer, 97-125.
- [31] Yüzkat, M., Sen, B., Caymaz, H. ve Karabiber, F., (2015). "Implementation of Data Generator for Process Mining Applications", SIU'2015, 1405-1408.
- [32] ProM Tools, <http://www.promtools.org/doku.php?id=tutorial:introduction>, 2015.
- [33] Process Mining, http://www.processmining.org/event_logs_and_models_used_in_book, 14 Temmuz 2015.
- [34] 3TU Datacentrum, http://data.3tu.nl/repository/collection:event_logs_real, 18 Ağustos 2015

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mecit YÜZKAT
Doğum Tarihi ve Yeri : 04.01.1988 - Van
Yabancı Dili : İngilizce
E-posta : muhmecit65@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Mühendislik	Trakya Üniversitesi	2012
Lise	Fen Bilimleri	Van Atatürk Lisesi	2006

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2013	Yıldız Teknik Üniversitesi	Araştırma Görevlisi
2012	Muş Alparslan Üniversitesi	Araştırma Görevlisi

YAYINLARI

Bildiri

1.Yüzkat, M., Sen, B., Caymaz, H. ve Karabiber, F., (2015)."Implementation of Data Generator for Process Mining Applications", SIU'2015, 1405-1408.