



SAKARYA
ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BULANIK MANTIK VE YAPAY SİNİR AĞLARINA GİRİŞ
1.ÖDEV RAPORU

B191210007 – Şevin Sena DERE

1/A GRUBU

DR. ÖĞR. ÜYESİ M. FATİH ADAK

ÖDEV KONUSU: Trafik ışığının yeşil yanma süresi; yeşil ışık yanan yoldaki araç sayısına, kırmızı ışıkta bekleyen araç sayısına bağlıdır

ÖDEVDE KONUYA AİT 2 DEĞİŞKEN VE BİR SONUÇ BULUNMAKTADIR.
DEĞİŞKEN 1: KIRMIZI DA BEKLEYEN ARAÇ SAYISI
DEĞİŞKEN 2: YEŞİL IŞIK YANAN YOLDAKİ ARAÇ SAYISI
SONUÇ: YEŞİL YANMA SÜRESİ (SN)

ÖDEVDE SIRASIYLA;
BULANIKLAŞTIRMA,
BULANIK SONUÇ ÇIKARMA,
DURULAMA İŞLEMLERİ YAPILMIŞTIR.

Bu ödevde, yeşil ışıktaki geçen ve kırmızı ışıktaki bekleyen araç yoğunluğu için, sıfır yoğunluk, düşük yoğunluk, orta yoğunluk ve yüksek yoğunluk kümeleri kullanılmıştır. Yeşil ve kırmızı ışık için kümelerin değer aralıkları farklılaşmıştır. Çünkü yeşil ışıktaki geçen ve kırmızı ışıktaki bekleyen araç mesafesi farklı olduğu için yoğunluk da farklı olur. Kırmızı ışıktaki araçlar hareket etmediği için aynı mesafe için daha fazla araç olacaktır. Yeşil ışıktaki araçlar hareket halinde olacağı için aralarındaki mesafe daha uzun olacaktır. Bu yüzden daha az araç olacaktır. Referans çalışmada 50 metre mesafe içerisinde farklı yoğunlukta araç olduğu örnek olarak seçilmiştir. Bu ödevde ise 100 metre mesafe içindeki araç sayısına göre değer aralıkları oluşturulmuştur. Çalışmaya göre, yeşil ışık için sıfır yoğunluk kümesi 0,1 araç, düşük yoğunluk kümesi 0, 1, 2, 3, 4, 5, 6 araç, orta yoğunluk kümesi 4, 5, 6, 7, 8 araç, yüksek yoğunluk kümesi 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 araç olarak belirlenmiştir. İlgili kümelere farklı derecelerde üye olmaktadırlar. Kırmızı ışık için sıfır yoğunluk kümesi 0,1 araç, düşük yoğunluk kümesi 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 araç, orta yoğunluk kümesi 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 araç, yüksek yoğunluk kümesi 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 araç olarak belirlenmiştir [1]. Çıkış fonksiyonu için süreler sıfır, kısa, kısa-normal, normal, normal-uzun, uzun, çok-uzun şeklinde belirlenmiştir. Çıkış fonksiyonuna ait değer aralıkları, sıfır süre kümesi 0-25 saniye aralığında, kısa süre kümesi 15-35 saniye aralığında, kısa-normal süre kümesi 25-45 saniye aralığında, normal süre kümesi 35-55 saniye aralığında, normal-uzun süre kümesi 45-65 saniye aralığında, uzun süre kümesi 55-90 saniye aralığında, çok-uzun süre kümesi 75-110 saniye aralığında olmaktadır.

Ödev mantığında kavşak modeli baz alınmıştır. Örneğin Doğu-Batı düzleminde lambalar kırmızı yanarken Kuzey-Güney düzleminde ise lambalar yeşil yanacaktır. Ya da tam tersi. Yoğunluklar tespit edildikten sonra bulanıklaştırma, çıkarım işlemleri, kural oluşturma ve durulama işlemleri yapılmıştır. Giriş ve çıkış değerlerine göre üyelik fonksiyonları oluşturulmuştur. Giriş değerlerinin her biri için 4 üyelik fonksiyonu bulunmaktadır. Buna uygun olarak grafik çizimi yapılmıştır.

Sistemin giriş büyüklükleri olan yeşil ışıktaki geçen araç sayısına ait görüntü Şekil 1’ de, üyelik fonksiyonu Şekil 1.1’ de, kırmızı ışıktaki bekleyen araç sayısına ait görüntü Şekil 2’ de, üyelik fonksiyonu Şekil 2.1’ de, çıkış büyüklüğü olan yeşil ışık yanma süresine ait görüntü Şekil 3’ de, üyelik fonksiyonu Şekil 3.1’ de gösterilmiştir. Kurallara göre çıktı üzerinde oluşturulan taralı alan ise Şekil 4’ de verilmiştir.

Sonraki aşamada kurallar oluşturulmuştur. If – Then ifadeleri kullanılarak ‘AND’ operatörüyle birleştirilerek formülize edilmiştir.

Örnek: IF yeşilde geçen araç sayısı IS düşük AND kırmızıda bekleyen araç sayısı IS sıfır THEN yeşilde bekleme süresi IS normal-uzundur.

Yeşil ışıktaki geçen araç sayısı üyelik fonksiyonu 4 tane, kırmızı ışıktaki bekleyen araç sayısı üyelik fonksiyonu 4 tane olduğu için $4 * 4 = 16$ tane kural kombinasyonu oluşur. Bununla beraber bazı kurallar gereksiz olacağı için bu sayı 13 kurala indirgenmiştir. Yeşil ışıktaki bekleyen araç sayısı sıfır olduğunda yeşil ışık yanma süresinin en kısa hali olması

gerektiği için yeşil ışık süresi, her yeşil araç sıfır kümesine dahil olduğunda yeşil yanma süresi de sıfır kümesinde olacaktır. Bu yüzden yeşil araç sayısı sıfır kümesini içeren diğer 3 kuralın yazılması gereksiz olacaktır. Kurallarda yeşil ışıkta geçen araç sayısı fazla, kırmızı ışıkta bekleyen araç sayısı sabit kalırsa yeşil yanma süresi artacaktır. Tam tersi yeşil ışıkta geçen araç sayısı az olup kırmızı ışıkta bekleyen araç sayısı fazla ise yeşil yanma süresi az azalacaktır. Eğer yeşil ve kırmızı ışıktaki araç sayısı yüksek ise yeşil yanma süresi normla süre kümesinde olacaktır. Kurallar bu çerçevede oluşturulmuştur. Kurallar toplu bir şekilde Şekil 5'te verilmiştir.

GİRİŞ-ÇIKIŞ ÜYELİK FONKSİYONLARI

```
VAR_INPUT
    yesilArac : REAL;
    kirmiziArac : REAL;
END_VAR
```

```
VAR_OUTPUT
    yesilSure : REAL;
END_VAR
```

```
FUZZIFY yesilArac
    TERM sifir := (0,1)(1,0);
    TERM dusuk := TRAPE 0 2 4 6;
    TERM orta := (4,0)(6,1)(8,0);
    TERM yuksek := TRAPE 6 8 24 24;
END_FUZZIFY
```

Şekil 1. yesilArac üyelik fonksiyonu için nitelik ve değer aralıkları

```
FUZZIFY kirmiziArac
    TERM sifir := (0,1)(1,0);
    TERM dusuk := TRAPE 0 2 10 12;
    TERM orta := (6,0)(12,1)(18,0);
    TERM yuksek := TRAPE 14 18 24 24;
END_FUZZIFY
```

Şekil 2. kirmiziArac üyelik fonksiyonu için nitelik ve değer aralıkları

```
DEFUZZIFY yesilSure
    TERM sifir := TRAPE 0 0 15 25;
    TERM kisa := (15,0)(25,1)(35,0);
    TERM kisaNormal := (25,0)(35,1)(45,0);
    TERM normal := (35,0)(45,1)(55,0);
    TERM normalUzun := (45,0)(55,1)(65,0);
    TERM uzun := (55,0)(65,1)(90,0);
    TERM cokUzun := TRAPE 75 90 110 110;
    METHOD : COG;
    DEFAULT := 30;
END_DEFUZZIFY
```

Şekil 3. yesilSure üyelik fonksiyonu için nitelik ve değer aralıkları

```

RULEBLOCK kuralBlok1
AND : MIN;
ACT : MIN;
ACCU : MAX;

RULE 1 : IF yesilArac IS sifir THEN yesilSure IS sifir;
RULE 2 : IF yesilArac IS dusuk AND kirmiziArac IS sifir THEN yesilSure IS normalUzun;
RULE 3 : IF yesilArac IS dusuk AND kirmiziArac IS dusuk THEN yesilSure IS normal;
RULE 4 : IF yesilArac IS dusuk AND kirmiziArac IS orta THEN yesilSure IS kısaNormal;
RULE 5 : IF yesilArac IS dusuk AND kirmiziArac IS yuksek THEN yesilSure IS kısa;
RULE 6 : IF yesilArac IS orta AND kirmiziArac IS sifir THEN yesilSure IS uzun;
RULE 7 : IF yesilArac IS orta AND kirmiziArac IS dusuk THEN yesilSure IS normalUzun;
RULE 8 : IF yesilArac IS orta AND kirmiziArac IS orta THEN yesilSure IS normal;
RULE 9 : IF yesilArac IS orta AND kirmiziArac IS yuksek THEN yesilSure IS kısa;
RULE 10 : IF yesilArac IS yuksek AND kirmiziArac IS sifir THEN yesilSure IS cokUzun;
RULE 11 : IF yesilArac IS yuksek AND kirmiziArac IS dusuk THEN yesilSure IS uzun;
RULE 12 : IF yesilArac IS yuksek AND kirmiziArac IS orta THEN yesilSure IS normalUzun;
RULE 13 : IF yesilArac IS yuksek AND kirmiziArac IS yuksek THEN yesilSure IS normal;

END_RULEBLOCK

```

Şekil 5-Kurallar tablosu

KODLAR VE ÇIKTILARI

Projede üyelik fonksiyonları tanımlama, bulanıklaştırma, durulum ve kural oluşturma işlemlerini yapabilmek için TrafikIsik.fcl uzantılı bir dosya oluşturuldu.

Üyelik fonksiyonları:

Girdiler:

-yesilArac

-kirmiziArac

Çıktı:

-yesilSure

```

package fuzzyLogic0dev1;

import java.io.File;

public class TrafikIsik {
    private final FIS fis;
    private double yesilArac;
    private double kirmiziArac;

    public TrafikIsik(double yesilArac, double kirmiziArac) throws URISyntaxException {
        this.yesilArac = yesilArac;
        this.kirmiziArac = kirmiziArac;

        File file = new File(getClass().getResource("TrafikIsik.fcl").toURI());
        fis = FIS.load(file.getPath(), true);

        fis.setVariable("yesilArac", yesilArac);
        fis.setVariable("kirmiziArac", kirmiziArac);
        fis.evaluate();
    }

    public double getSure(double yesilArac, double kirmiziArac) {
        fis.setVariable("yesilArac", yesilArac);
        fis.setVariable("kirmiziArac", kirmiziArac);
        fis.evaluate();
        return Math.round(fis.getVariable("yesilSure").getValue());
    }

    public void grafikCiz() {
        JFuzzyChart.get().chart(fis);
        JFuzzyChart.get().chart(fis.getVariable("yesilSure").getDefuzzifier(), "Yeşil Yama Süresi", true);
    }

    public String toString() {
        return "Yeşil Işık Yama Süresi(sn): " + Math.round(fis.getVariable("yesilSure").getValue()) + " sn";
    }

    public FIS getModel() {
        return fis;
    }
}

```

```

package fuzzyLogic0dev1;

import java.io.IOException;

public class Program {

    public static void main(String[] args) throws URISyntaxException, IOException {
        Scanner in = new Scanner(System.in);
        double yesilArac, kirmiziArac;

        do {
            System.out.print("Yeşil Işık Yanan Yoldaki Araç Sayısı : ");
            yesilArac = in.nextDouble();
            System.out.print("Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : ");
            kirmiziArac = in.nextDouble();
        } while (yesilArac > 24 || kirmiziArac > 24);

        TrafikIsik trafikIsik = new TrafikIsik(yesilArac, kirmiziArac);

        try {
            var kurallar = trafikIsik.getModel().getFunctionBlock("TrafikIsik").getFuzzyRuleBlock("kuralBlok1").getRules();
            for (var kural : kurallar) {
                if (kural.getDegreeOfSupport() > 0) {
                    System.out.println(kural);
                }
            }
            System.out.println("Yeşil Işık Yama Süresi(sn) : " + trafikIsik.getSure(yesilArac, kirmiziArac) + " sn");
            JFuzzyChart.get().chart(trafikIsik.getModel());
            JFuzzyChart.get().chart(trafikIsik.getModel().getVariable("yesilSure").getDefuzzifier(), "Yeşil Yama Süresi", true);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Program Dosyası Görüntüsü

TrafikIsik Class Görüntüsü

```

Problems @ Javadoc Declaration Console X
Program (3) [Java Application] /Users/sevinsenedere/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jr
Yeşil Işık Yanan Yoldaki Araç Sayısı : 25
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 28
Yeşil Işık Yanan Yoldaki Araç Sayısı : 10
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 15
12 (0.5) if (yesilArac IS yuksek) AND (kirmiziArac IS orta) then yesilSure IS normalUzun [weight: 1.0]
13 (0.25) if (yesilArac IS yuksek) AND (kirmiziArac IS yuksek) then yesilSure IS normal [weight: 1.0]
Yeşil Işık Yama Süresi(sn) : 52.0 sn

```

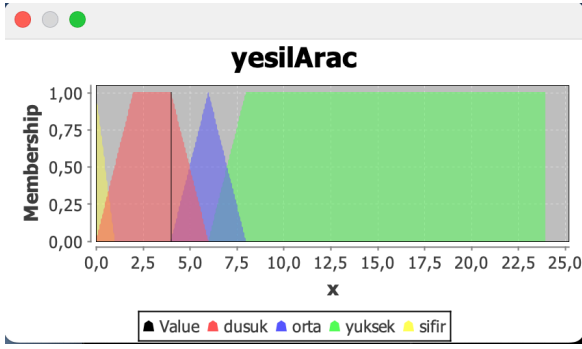
Programı çalıştırdığımızda yeşil ve kırmızı ışıktaki araç sayısını girmemizi isteyen ekran karşımıza gelir. Eğer biz 24 üstünde bir değer girersek değerleri tekrar girmemiz gerekecektir. Aralık içerisinde girdiler verildiğinde program çalışıp doğru sonuçlar vermektedir.

Örneğin yeşil ışık için 4, kırmızı ışık için 9 değerlerini girdiğimizde bize 41 saniye gibi bir yeşil ışık yanma süresi verir. 4 yeşilArac kümesinde düşük değer aralığındadır. 9 ise kırmızıArac kümesinde hem düşük hem orta değer aralığındadır. Bu yüzden 2×1 'den 2 kural çalışır. yeşilArac düşük ve kırmızıArac düşük(3.kural), yeşilArac düşük ve kırmızıArac orta (4.kural) kuralları çalışır. 3.kural çıktısı normal, 4.kural çıktısı kısaNormal olmaktadır. 3.kuralda düşük ve düşük tablolarında minimum değer olan tablo normal tablo çıktısında taranır. 4.kuralda düşük ve orta tablolarında minimum değer olan tablo kısaNormal tablo çıktısında taranır. Sonra bu taralı alanlar birleştirilerek Şekil 4'teki çıktı elde edilir.

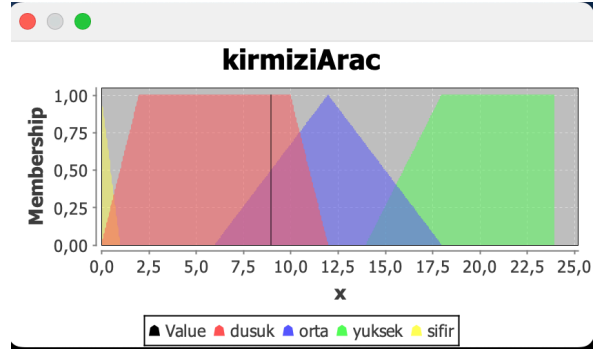
```

Problems Javadoc Declaration Console X
Program (3) [Java Application] /Users/sevinsenadere/.p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jre/b
Yeşil Işık Yanan Yoldaki Araç Sayısı : 4
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 9
3 (1.0) if (yeşilArac IS düşük) AND (kırmızıArac IS düşük) then yeşilSure IS normal [weight: 1.0]
4 (0.5) if (yeşilArac IS düşük) AND (kırmızıArac IS orta) then yeşilSure IS kısaNormal [weight: 1.0]
Yeşil Işık Yanma Süresi(sn) : 41.0 sn

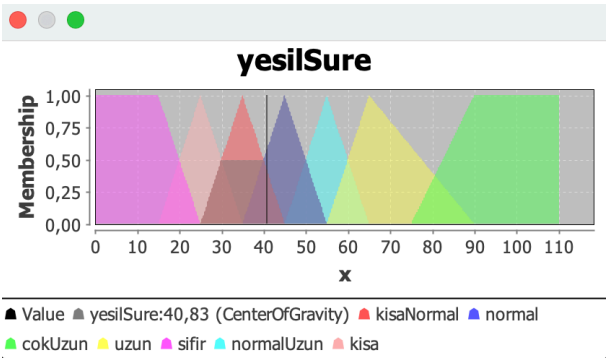
```



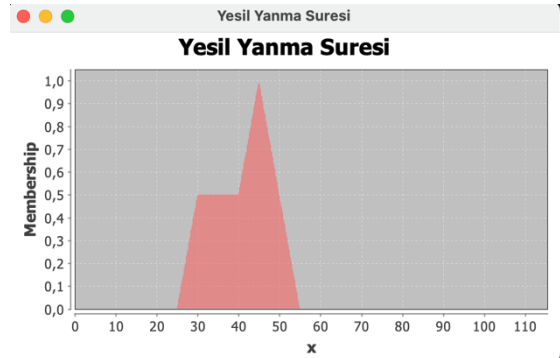
Şekil 1.1



Şekil 2.1



Şekil 3.1

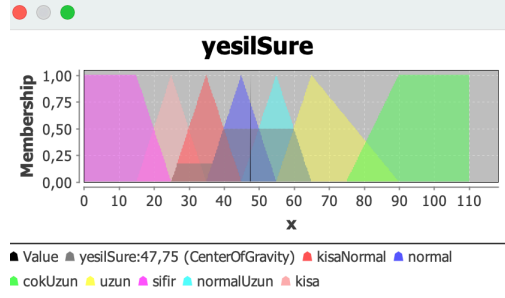


Şekil 4

-COG DURULAMA METODUNA GÖRE GARFİK ÇIKTISI

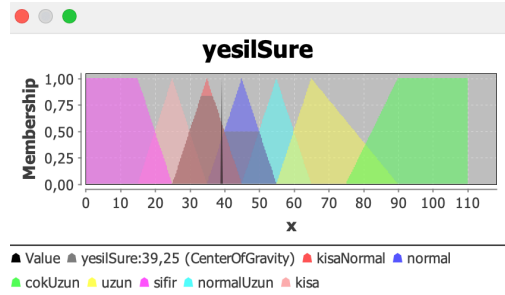
□ 1.ÖRNEK

```
Problems Javadoc Declaration Console X
Program (3) [Java Application] /Users/sevinsnadere/p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jre/bin/java (17 Kas
Yeşil Işık Yanan Yoldaki Araç Sayısı : 5
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 7
3 (0.5) if (yesilArac IS dusuk) AND (kirmiziArac IS dusuk) then yesilSure IS normal [weight: 1.0]
4 (0.16666666666666666) if (yesilArac IS dusuk) AND (kirmiziArac IS orta) then yesilSure IS kısaNormal [weight: 1.0]
7 (0.5) if (yesilArac IS orta) AND (kirmiziArac IS dusuk) then yesilSure IS normalUzun [weight: 1.0]
8 (0.16666666666666666) if (yesilArac IS orta) AND (kirmiziArac IS orta) then yesilSure IS normal [weight: 1.0]
Yeşil Işık Yanma Süresi(sn) : 48.0 sn
```



□ 2.ÖRNEK

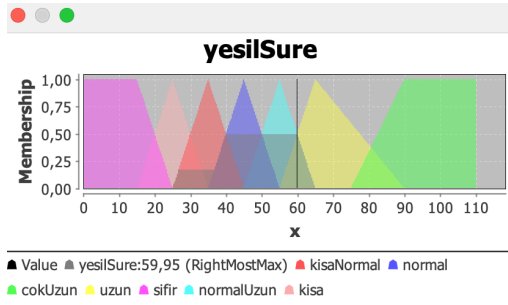
```
Problems Javadoc Declaration Console X
Program (3) [Java Application] /Users/sevinsnadere/p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jre/bin/java (17 Kas 2
Yeşil Işık Yanan Yoldaki Araç Sayısı : 3
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 11
3 (0.5) if (yesilArac IS dusuk) AND (kirmiziArac IS dusuk) then yesilSure IS normal [weight: 1.0]
4 (0.8333333333333334) if (yesilArac IS dusuk) AND (kirmiziArac IS orta) then yesilSure IS kısaNormal [weight: 1.0]
Yeşil Işık Yanma Süresi(sn) : 39.0 sn
```



-RM DURULAMA METODUNA GÖRE GRAFİK ÇIKTISI

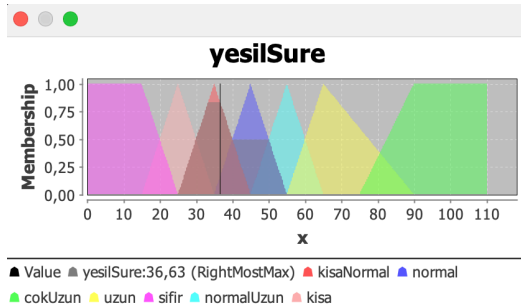
□ 1.ÖRNEK

```
Problems Javadoc Declaration Console X
Program (3) [Java Application] /Users/sevinsnadere/p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jre/bin/java (17 Kas
Yeşil Işık Yanan Yoldaki Araç Sayısı : 5
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 7
3 (0.5) if (yesilArac IS dusuk) AND (kirmiziArac IS dusuk) then yesilSure IS normal [weight: 1.0]
4 (0.16666666666666666) if (yesilArac IS dusuk) AND (kirmiziArac IS orta) then yesilSure IS kısaNormal [weight: 1.0]
7 (0.5) if (yesilArac IS orta) AND (kirmiziArac IS dusuk) then yesilSure IS normalUzun [weight: 1.0]
8 (0.16666666666666666) if (yesilArac IS orta) AND (kirmiziArac IS orta) then yesilSure IS normal [weight: 1.0]
Yeşil Işık Yanma Süresi(sn) : 60.0 sn
```



2.ÖRNEK

```
Program (3) [Java Application] /Users/sevinsenedere/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_20.0.2.v20230801-2057/jre/bin/java (17 Kas :
Yeşil Işık Yanan Yoldaki Araç Sayısı : 3
Kırmızı Işık Yanan Yolda Bekleyen Araç Sayısı : 11
3 (0.5) if (yesilArac IS dusuk) AND (kirmiziArac IS dusuk) then yesilSure IS normal [weight: 1.0]
4 (0.8333333333333334) if (yesilArac IS dusuk) AND (kirmiziArac IS orta) then yesilSure IS kisaNormal [weight: 1.0]
Yeşil Işık Yanma Süresi(sn) : 37.0 sn
```



İlk örnekte 5 ve 7 değerleri verilir. COG metoduna göre 48 saniye yanma süresi verilmiştir. Taralı alan oluştuktan sonra ağırlık merkezi bulunur. RM metoduna göre aynı örnekte 60 saniye yanma süresi vermiştir. RM kullandığımızda değerin sağa kaydığını gözlemledik. Değerleri azaltsak bile RM ile sürekli taralı alanın maksimum noktasındaki değeri verir. Bir değişime uğramaz. Bu metodun kullanımı ışığın fazladan yanmasına sebep olabilir. Yani gereksiz bir bekleme süresi demek olur. İkinci örnekte 3 ve 11 değerleri verilir. COG metoduna göre 39 saniye yanma süresi vermiştir. RM metoduna göre ise 37 saniye vermiştir. Bunun sebebi maksimum noktanın sol kümede bulunmasıdır. Sol kümenin maksimum noktasını getirir. Değerler artırıldığında COG metodunda ağırlık merkezi değişmektedir. RM metodunda ise maksimum noktası değişmediği sürece yeşil ışık yanma süresinde bir değişim olmayacaktır. Yeşil ışık yanma süresi araç sayılarına bağlı olduğu için sabit kalmasını istemeyiz. Bu örnek için RM metodu iyi bir durulama yöntemi değildir.

KAYNAKÇA:

[1] ERDEM, O. 2007 “KAVŞAK TRAFİK SİNYALİZASYON SİSTEMİ İÇİN BULANIK MANTIK TABANLI GERÇEK ZAMANLI DENETLEYİCİ TASARIMI VE UYGULAMASI”, e-Journal of New World Sciences Academy, (241-255. s.)