

HALI FABRİKASINDA ENAZ BOBİN DEĞİŞTİRMEYİ SAĞLAYACAK ÜRETİM SİRALAMASININ BELİRLENMESİ

Yrd.Doç.Dr. Filiz ÇALIŞKAN*

GİRİŞ

Atlantik Halı fabrikasında, farklı renk bileşimlerinden oluşturulmuş yaklaşık 200 halı modeli üretilmektedir. Dokuma bölümündeki tezgahların her birinde ise maksimum 5'er iplik bobini bulunmaktadır. Halı üretiminde, her bir bobine değişik renkte iplik takılmakta ve renkleri belirlenmiş model işlenmektedir. Farklı modellerin işlenmesi için bobin değişimi sıklıkla gerçekleştirilmekte ve büyük maliyetlere neden olmaktadır. Dolayısı ile en az değişimi sağlayacak model sıralamasının belirlenmesi, üretim programının oluşturulması açısından önem kazanmaktadır. Şimdiye kadar tecrübeye dayalı sezgisel yaklaşımla gerçekleştirilen üretim programının, bilgisayar destekli optimizasyonu yönetim tarafından arzulanmaktadır. Söz konusu amaç doğrultusunda gerçekleştirilen bu çalışmada, önce tezgah üzerinde eniyi renk sıralaması bulunacak, sonra kullanılan renkler temelinde yakınlık sağlayan modeller eşleme biçiminde kıyaslanarak, birbirleri arasında renk çakışma ve bobin değiştirme sayıları oluşturulacaktır. Ardısıra, elde edilen çakışma ve değiştirme matrisleri yardımı ile enaz değiştirmeyi gerçekleştirecek model sıralaması bulunacaktır.

1. PROBLEMİN TANIMLANMASI

İşletmenin üretime başlaması esnasında iki farklı başlangıç durumu sözkonusu olabilir. Birincisi, tezgahlar boştur ve üretime yeni başlanacaktır. Yani, bobinlere henüz farklı renklerde iplikler takılmamıştır. Bu durumda, doğal olarak üretimden sorumlu kişi ard arda üretmeyi planladığı modeller içerisinde en çok kullanılan renkteki iplikleri bobinlere takacaktır. Bu varsayımdan hareketle çalışmada öncelikle tezgah üzerinde eniyi renk sıralaması bulunacak, sonra kullanılan renkler temelinde yakınlık sağlayan modeller eşleme biçiminde kıyaslanarak, birbirleri arasında renk çakışma ve bobin değiştirme sayıları oluşturulacaktır. Ardısıra, elde edilen çakışma ve değiştirme matrisleri yardımı ile enaz değiştirmeyi gerçekleştirecek model sıralaması bulunacaktır. İkincisi, tezgahta bir model üretimi tamamlanmış ve bobinlerde, o modelde gerekli renkteki iplikler bulunmaktadır. Bu durumda ise bobinlerde bulunan farklı renkteki iplikler başlangıç noktası olarak belirlenecek ve ardından, kullanılan renkler temelinde yakınlık sağlayan modeller eşleme biçiminde kıyaslanarak, birbirleri arasında renk çakışma ve bobin değiştirme sayıları oluşturulacaktır. Bu işlemin ardısıra yine, elde edilen çakışma ve değiştirme matrisleri yardımı ile enaz değiştirmeyi gerçekleştirecek model sıralaması belirlenecektir. Çalışmada gerçek bir probleme çözüm getirmek amaçlandığı ve çözüm sürecinin başlaması için mutlaka bir başlangıç noktası gerektiği için her iki durumda gözönüne alınmıştır. Çözüm süreci açısından sözkonusu iki durum arasındaki tek fark, ilkinde başlangıç renklerinin geliştirilen bir alt program yardımı ile belirlenmesine karşılık, ikincisinde başlangıç renkleri olarak tezgahta en son dokunmuş olan modelin renklerinin veri olarak alınmasıdır. Bobin değiştirme sayısının minimum olması amacı doğrultusunda bu fark çözüm sürecinde önemli bir değişiklik gerektirmezken optimum çözümlü etkilemektedir. Başlangıç renkleri olarak tezgahta en son dokunmuş olan modelin

* Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Öğretim Üyesi.

renklerinin veri olarak alınması bobin deęiřtirme sayısını artırabilmektedir. Bu artış üretilecek model sayısının büyümesi durumunda daha açık olarak izlenmektedir. Her iki durumda getirilecek çözümde izlenecek yol başlangıç renklerinin belirlenmesinden sonra aynen olacağı için çalışma kapsamında sadece ilk durum bir örnek üzerinde açıklanacaktır. Yöntemin çalışmasının anlaşılabilmesi amacı ile örnekteki model sayısı oldukça küçük tutulmuştur.

Problemin çözümü amacı ile geliştirilen modelde bobin deęiřtirme maliyetleri de gözönüne alınmakla birlikte bu konuda işletmeden gerekli ve yeterli bilgi alınmadığı için her bobin deęiřtirmenin maliyeti eşit olarak kabul edilmiş ve bu büyüklük 1 ile ifade edilmiştir.

2. DEęIŐKENLERİN TANIMLANMASI

$u = \{ u \mid u = 1, 2, 3, \dots, m \}$, renkler kümesi olarak kabul edildiğinde, her bir modelde bobin sayısı kadar renk tanımlaması bulunur.

$i = \{ i \mid i = 1, 2, 3, \dots, n \}$, modeller kümesi iken, atanacak yer-sıra sayısı da n kadardır. Böylece $j=i$ alınarak modeller ve alternatif sıraları için i ve j indisleri kullanılır. Modellerin atanabileceęi alternatif sıra seçeneęi t adet olup seçenekler kümesi ařaęıdaki gibidir.

$$S = \{ s \mid s = 1, 2, 3, \dots, t \}$$

d_{ij}^s : s seçeneęinde i ve j modelleri arasındaki bobin deęiřtirme maliyeti, (=1 durumu, sabit maliyet)

a_{ij}^s : s seçeneęinde i ve j modelleri arasındaki bobin deęiřtirme sayısı,

y_{ij}^s : s seçeneęinde i ve j modelleri arasındaki ortak renk sayısı (yakınlık deęeri),

C_s : s seçeneęinde toplam deęiřtirme sayısı,

K_s : s seçeneęinde toplam çakıřma deęeri,

C_{enk} : enküçük toplam bobin deęiřtirme sayısı,

$$C_{enk} = \text{Enk} \{ C_s \}$$

K_{enb} : Enbüyük toplam çakıřma deęeri,

$$K_{enb} = \text{Enb} \{ K_s \}$$

Tanımlanan deęiřkenler yardımı ile oluřturulan alternatif düzenlemeler uygun çözüm alanı olarak ele alındığında ve bu alanda t kadar seçenek olduęu kabul edildiğinde; alternatifler içinde enküçük deęiřtirme sayısına karřı gelen seçenek,

$$C_{enk} = \text{Enk} \sum_{i=1}^n \sum_{j=i+1}^n a_{ij}^s d_{ij}^s \quad (d_{ij}^s = 1 \text{ olduęunda}) \quad (1)$$

eřitlięiyle; ele alınan alternatifler içinde enbüyük çakıřma deęerine karřı gelen seçenek ise,

$$K_{enb} = Enb \sum_{i=1}^n \sum_{j=i+1}^n y_{ij}^s d_{ij}^s \quad (d_{ij}^s = 1 \text{ olduğunda}) \quad (2)$$

eşitliği ile bulunacaktır.

Çakışma durumu, değiştirme durumunun tersi olduğundan (1) ve (2) formülasyonlarından herhangi birisi kullanılabilir.

3. ÇÖZÜM ALGORİTMASI

Problemin çözümüne yönelik olarak geliştirilen algoritma aşağıdaki gibidir.

Adım 1. Model tanımları, tezgah sayısı, bir tezgahtaki bobin sayısı girdi olarak verilir.

Adım 2. **EniyiRenkSıra[modeller]** altprogramı ile tüm modellerde bobin sırası üzerinde en yoğun renk kullanımı (sıralaması) bulunur.

Adım 3. Adım 2 sonucu temelinde, **UygunModelSıra[modeller]** altprogramı ile başlangıç modeli için renk sıralaması geliştirilir.

Adım 4. **DeğişmeMat[modeller]** altprogramı ile her bir model bir diğeri ile değiştirildiğinde (ardından üretime alındığında) değiştirilecek bobin sayıları matrisi ve çakışan renk sayıları matrisi oluşturulur.

Adım 5. Yeri değiştirilmeyecek bobin numarası, modeller arası renk çakışma matrisi ya da modeller arası bobin değiştirme matrisi girdi olarak verilir. **EnAzDeğişme[çakışma]** altprogramı ile düzenlemenin oluşturduğu toplam değiştirme sayısı hesaplanır.

Adım 6. Her bir bobin, sırasıyla diğerlerinin konumu ile değiştirilerek değişme azaltımı (çakışma artırımı) sağlayan sıralama gerçekleştirilir.

Adım 7. Son düzenleme temelinde 6. adım, değişme azaltımı (çakışma artırımı) sağlanamayınca dek tekrarlanır.

Adım 8. Eniyi model sıralaması listelenir.

4. ÇÖZÜM ALGORİTMASININ ÖRNEK ÜZERİNDE AÇIKLAMASI

Algoritma adımları örnek üzerinde açıklanacaktır. Sektördeki model geliştirmedeki renk kullanımının belirtilmesi rekabetçi ortam koşullarında sakınca yaratacağından, renkler alfabetik olarak kodlanmış ve yoğun talep içeren ilk 5 model kullanılmıştır.

Adım 1. Model tanımları, tezgah sayısı, bir tezgahtaki bobin sayısı girdi olarak verilir.

modeller=(*Her bir model için tek satır*)

{a,f,c,d,e},

{a,c,b,d,e},

{d,b,c,a,e},
{a,b,c,d,e},
{c,b,e,d,a});

modeladlar={selale, rahvan, erciyes, fantasy, tahrان};

Adım 2. EniyiRenkSıra[modeller] altprogramı ile tüm modellerde bobin sırası temelinde renk kullanımı sıklığı bulunur ve sıralanır. (Her sütun için örnekteki beş modelde a renginden üç, b renginden üç, c renginden üç, d renginden dört ve e renginden dört adet bulunduğu belirlenmiştir.)

AynıRenkKullanımı[Transpose[modeller]]

{{3,a}, {3,b}, {3,c},{4,d}, {4,e}}

EniyiRenkSıra[modeller]

{a, b, c, d, e}

Adım 3. Adım 2 sonucu (sıra) temelinde, **Esleme[sıra,modeller]** altprogramı ile başlangıç model sıralaması geliştirilir. En fazla kullanılan renklerin bulunduğu başlangıç modeli ile diğer modeller arasındaki ilişki 1 ve 0 ile gösterilmiştir. 1'ler çakışan (aynı) renkleri, 0'lar ise değişmesi gereken renkleri göstermektedirler. Başlangıç model sıralaması çakışma sayısına göre düzenlenmiştir.

UygunModelSıra[modeller]

{{{1, 1, 1, 1, 1}, fantasy},
{{1, 0, 1, 1, 1}, selale},
{{1, 0, 0, 1, 1}, rahvan},
{{0, 1, 1, 0, 1}, erciyes},
{{0, 1, 0, 1, 0}, tahrان}}
{{5, fantasy}, {4, selale}, {3, rahvan}, {3, erciyes}, {2, tahrان}}

Model sırası (Alfabetik Kod):

{{fantasy, selale, rahvan, erciyes, tahrان}}

Model sırası (Sayısal Kod):

{4, 1, 2, 3, 5}

Adım 4. DegismeMat[modeller] altprogramı ile her bir model bir diğeri ile değiştirildiğinde (ardından üretime alındığında) değiştirilecek bobin sayıları matrisi ve çakışan renk sayıları matrisi oluşturulur.

DegismeMat[modeller]

degisme=

0	2	3	1	4
2	0	4	2	4
3	4	0	2	4
1	2	2	0	3
4	4	4	3	0

cakisma=

0	3	2	4	1
3	0	1	3	1
2	1	0	3	1
4	3	3	0	2
1	1	1	2	0

Adım 5. Yeri değiştirilmeyecek bobin numarası, modeller arası renk çakışma matrisi ya da modeller arası bobin değiştirme matrisi girdi olarak verilir. **EnAzDegisme[cakisma]** altprogramı ile önce başlangıç düzenlemenin oluşturduğu toplam değiştirme sayısı hesaplanır.

EnAzDegisme[cakisma]

BaşlangıçDeğer= 11 Sıralama : {4, 1, 2, 3, 5}

Adım 6. Her bir renk (bobin), sırasıyla diğerlerinin konumu ile değiştirilerek değişme azaltımı (çakışma artırımı) sağlayan sıralama gerçekleştirilir.

{2, 2, 3, 4} = 11 * Sıralama : {4, 2, 1, 3, 5}

{2, 2, 2, 4} = 10 * Sıralama : {1, 2, 4, 3, 5}

Adım 7. Son düzenleme temelinde 6. adım, değişme azaltımı (çakışma artırımı) sağlanamayınca dek yinelenir.

{2, 1, 2, 4} = 9 * Sıralama : {2, 1, 4, 3, 5}

Adım 8. Eniyi model sıralaması listelenir.

{2, 1, 2, 4} = 9 * Sıralama : {2, 1, 4, 3, 5}

ENİYİ ÇÖZÜM = {rahvan, selale, fantasy, erciyes, tahran}

Burada;

{2, 3, 2, 3} = 10 * Sıralama : {2, 1, 3, 4, 5} anlatımı;

2->1 için (2. modelden 1. modele geçmek için) 2 bobin değiştirilmelidir.

1->3 için (1. modelden 3. modele geçmek için) 3 bobin değiştirilmelidir.

3->4 için (3. modelden 4. modele geçmek için) 2 bobin değiştirilmelidir.

4->5 için (4. modelden 5. modele geçmek için) 3 bobin değiştirilmelidir.

Dolayısı ile {2,1,3,4,5} sırası toplam 10 bobin değiştirmeyi gerektirmektedir. Enaz değişiklik gerektiren sıralama ise 9 değiştirme değeri ile {2,1,3,4,5} olmaktadır. Baka bir deyişle, üretim programı; {rahvan, selale, fantasy, erciyes, tahran} biçiminde olmalıdır.

Eğer şu anda tezgahlar bir model üzerinde çalışıyor ise bundan sonra üretime alınması gerekli model türlerinin belirlenmesi için Sabit Konum opsiyonu kullanılarak çözüme gidilir. Örneğin, 3 nolu modelin (erciyes) işlendiği varsayalım. Bu durumda eniyi üretim programı izleyen biçimde bulunur:

EnAzDeğistirme[DegismeMat[modeller],SabitKonum->{3}]

BaşlangıçDeğer= 10 Sıralama : {3, 2, 1, 4, 5}

{2, 1, 2, 4} = 9 * Sıralama : {3, 4, 1, 2, 5}

ENİYİ ÇÖZÜM = {3, 4, 1, 2, 5}

{erciyes, fantasy, selale, rahvan, tahrان}

SONUÇ

Atlantik halı fabrikasında gözlenen yüksek bobin değiştirme sayısının neden olduğu maliyet problemi, aynı sektördeki benzer teknoloji kullanan tüm işletmelerde günceldir. Bu soruna çözüm getirmek üzere Mathematica dilinde yazılan bir program yardımı ile algoritması verilen bir yöntem önerilmiştir. Önerilen yöntem farklı örnek büyüklüklerinde de denenmiştir. Yöntemin temeli bir başlangıç bobin değiştirme sayısına ulaştıktan sonra tüm alternatif değişimleri inceleyerek en iyi değişim sayısını veren model üretim sırasını önerme esasına dayanmaktadır.

Çalışma kapsamında seçilen 5 farklı modelde alternatif model sıralama sayısı $5!=120$ dir. Yöntem çalışırken kendi içinde kısmi sayım yaklaşımını benimseyerek daha iyi bir çözüm buldukça, iyi olmayanları elimine etmektedir. Örneğimizde en çok kullanılan renk temelinde belirlenen model sıralamasında en az bobin değiştirme sayısı 11 iken, program daha düşük değiştirme sayısı bulmak amacı ile alternatifler üretmiş ve bobin değiştirme sayısı 9 olan **rahvan, selale, fantasy, erciyes, tahrان** model sıralamasına ulaşmıştır.

Çalışmada, sabit konum durumu yani, bobinlerde en son üretilen modelin iplik renklerinin bulunması bir örnek çözüm ile gösterilmiştir. Verilen örnekte model erciyes d,b,c,a,e renk kodları ile başlangıç modeli olarak varsayılmıştır. Ulaşılan en iyi bobin değiştirme sayısı yine 9 bulunurken model sırası **erciyes, fantasy, selale, rahvan, tahrان** olarak önerilmiştir. Görüldüğü gibi aynı bobin sayısını veren farklı model sırası olabilmektedir. Bu durum işletmenin en az bobin değiştirmek kaydı ile üretimde herhangi bir modele öncelik vermesine de imkan sağlaması açısından olumlu bulunmaktadır. Özellikle siparişe göre çalışan işletmelerde bu bir avantaj olarak yorumlanabilir. Talebin fazla olduğu veya sipariş süresi açısından öncelik verilmesi gereken bir model var ise, öncelikle o modelin ilk sırada bulunduğu sıralama alternatifler içinden seçilebilir. Şayet o model önerilen alternatif sıralamaların hiç birisinde ilk veya ilke yakın sırada bulunmuyor ise o zaman sabit konumlu başlangıç durumunda çalışılması önerilir. Programın 20 örnek ile çalıştırılmasında ise başlangıç durumu en iyi renk sıralaması temelinde alındığında 27 bobin değiştirme en iyi sayı iken, sabit konumda bir modelin seçilmesi sonrasında bobin değiştirme sayısı 29'a ulaşmıştır.

İncelenen problemde tüm halılarda kullanılan renk sayısının beş olduğu kabul edilmiştir. Yapılan araştırma sonucunda ise üretilen modellerde kullanılan maksimum renk sayısının da beş olduğu bilinmektedir. Ancak, farklı modellerde bu renk sayısı beş veya beşten az olmaktadır. Yani, tek renk iplikte üretilen modeller de bulunmaktadır. Önerilen yön-

temin örnek üzerinde gösteriminde bu durum göz önüne alınmamaktadır ve zaaflık olarak yorumlanabilir. Böyle bir durumda yani, modellerdeki renk sayısının farklı olması durumunda, yapılacak tek işlem beşten az olan her modelin renk tanımlanmasında renk sayısını, renklerin tanımlanmasında kullanılmayan bir kod (harf) ile beşe tamamlamaktır. Önerilen bilgisayar programında hiç bir değişiklik gerekmemektedir.

EKLER. Algoritmanın Mathematica dili ile kodlanması.

```

ToplamDegistirme[list__List]:=
Block[{yol1=list},
  dege={};
  For[x=1,x!=Length[yol1]+1,x++,
    al=yol1[[x]];
    ilk1=al[[1]];
    ilk2=al[[2]];
    deg=enk[[ilk1,ilk2]];
    dege=Append[dege,deg];
  ];deger=Apply[Plus,dege]
]

```

```

Frequencies[list_List] :=
  Map[ {#, Count[list, #]}&, Union[list] ]
ColumnTake[data:{__List}, spec_] :=
  Transpose[Take[Transpose[data], spec]]
CumulativeSums[list_List] := Accumulate[Plus, list]
SquareMatrixQ[mat_?MatrixQ]:=
  MatrixQ[mat] && (Length[mat]==Length[Transpose[mat]])

```

```

Attributes[Repeat]={HoldAll}
Repeat[body_,test_]:=While[True,body;If[test,Break[] ] ]

```

```

AynıModelKullanımı[list_List] :=
  Map[ {#, Count[list, #]}&, Union[list] ]

```

```

AynıRenkKullanımı[list_List] :=

```

```

  Table[
    st=Transpose[AynıModelKullanımı[list[[s]]]];
    enb=Max[st[[2]]];
    yer=Position[st[[2]],enb][[1,1]];
    st2={enb,st[[1,yer]]}
    ,{s,Length[list]} ]

```

```

EniyiRenkSıra[list__]:=

```

```

Block[{modeller=list},
  Transpose[AynıRenkKullanımı[Transpose[modeller]]][[2]]
]

```

```

Esleme[dat__,data__]:=

```

```

Block[{set=dat,modeller=data,x,x1},

```

```

        modelad=modeladlar;
        esdata={};
For [x=1,x!=Length[modeller]+1,x++,
    model=modeller[[x]];es={};ad=modelad[[x]];
For [x1=1,x1!=Length[model]+1,x1++,
    If[set[[x1]]==model[[x1]],
        es=Append[es,1],,es=Append[es,0]
    ]
];esdata=Append[esdata,{es,ad}];
];
eniye=Reverse[Sort[esdata]]
]

```

```

UygunModelSira[datafil__] :=
Block[{modeller=datafil},
numtotext={};
Do[tt=modeladlar[[jwx]]->jwx;
numtotext=Append[numtotext,tt],
{jwx,Length[modeladlar]};
texttonum={};
Do[tt=jwx->modeladlar[[jwx]];
texttonum=Append[texttonum,tt],
{jwx,Length[modeladlar]};
numadlar=adlar/.numtotext;
numsetler=setler/.numtotext;
al=EniyiRenkSira[modeller];
bul=Esleme[al,modeller];
fil=Transpose[bul];
fil2=fil[[2]];fill=fil[[1]];
top={};
Do[gg=Apply[Plus,fill[[ee]]];
top=Append[top,gg],
{ee,1,Length[fill]};
toplaml={top,fil2};
toplaml=Transpose[toplaml];
tamam=Reverse[Sort[toplaml]];
ensira=Transpose[tamam][[2]]
]

```

```

DeğismeMat[list5__List] :=

```

```

Block[{modeller=list5,jwx,y,i,j,i11,j11,z,i7,p9},
    cakıs={};adlar={};setler={};
    modelad=modeladlar;
For [y=1,y!=Length[modeller]+1,y++,
    set=modeller[[y]];ad1=modeladlar[[y]];
    setler=Append[setler,ad1];
    aldatam=Esleme[set,modeller];
    aldata=Drop[aldatam,1];
    adsıra=Transpose[aldata][[2]];
    data=Transpose[aldata][[1]];
    adata=Transpose[data];
    satoplaml=Last[CumulativeSums[adata]];

```



```

cakıs=Append[cakıs,satoplam];
adlar=Append[adlar,adsıra];
];
numtotext={};
Do[tt=modeladlar[[jwx]]->jwx;
numtotext=Append[numtotext,tt],
{jwx,Length[modeladlar]};
texttonum={};
Do[tt=jwx->modeladlar[[jwx]];
texttonum=Append[texttonum,tt],
{jwx,Length[modeladlar]};
numadlar=adlar/.numtotext;
numsetler=setler/.numtotext;

(*sıfırlama*)

bostablo=
Table[{i11,j11},{i11,1,Length[modeler]},
{j11,1,Length[modeler]};
cakısma=bostablo;
For[z=1,z!=Length[cakıs]+1,z++,
value=cakıs[[z]];
yap=Length[cakıs]-1;
ov=numsetler[[z]];
ilk=Table[ov,{yap}];
iki=numadlar[[z]];
koord=Transpose[{ilk,iki}];
bb11={};
Do[ttt=koord[[it]]->value[[it]];
bb11=Append[bb11,ttt],
{it,Length[koord]};
cakısma=cakısma/.bb11;
];
sfr=Transpose[{numsetler,numsetler}];
bb2={};
Do[ttt1=sfr[[p9]]->0;
bb2=Append[bb2,ttt1],
{p9,Length[sfr]};
cakısma=cakısma/.bb2;
enk=Map[If[#>0,5-#,#]&,cakısma,{2}]
]

Options[EnAzDegistirme]={ SabitKonum -> {None} }

EnAzDegistirme[coordmat_,opts___Rule]:=
Block[
{degmat=coordmat},
alt=Table[km,{km,1,Length[degmat]};
sabityer=SabitKonum /.{opts} /.Options[EnAzDegistirme];

If[sabityer!={None},
alt1=alt;

```

```

alt2=alt1;
sabityer=Flatten[sabityer,1][[1]];
sab=Position[alt,sabityer][[1]];
alt1[[1]]=alt2[[sab]];
alt1[[sab]]=alt2[[1]];
alt=alt1;
yol=Partition[alt,2,1];
minval=ToplamDegistirme[yol];
Print["BaşlangıçDeğer= ",minval," Sıralama : ",alt];
Repeat[
  RatInc=False;
  For[i=1,i!=Length[degmat],i++,
    If[!MemberQ[{sabityer},i],
  For[j=i+1,j!=Length[degmat]+1,j++,
    If[!MemberQ[{sabityer},j],
      (*Print["i= ",i," : j= ",j];*)
      ver1=alt;
      ver2=ver1;
      ii=Position[alt,i][[1,1]];
      jj=Position[alt,j][[1,1]];
      ver1[[ii]]=ver2[[jj]];
      ver1[[jj]]=ver2[[ii]];
      yol=Partition[ver1,2,1];
      minvall=ToplamDegistirme[yol];
      If[minvall < minval,
        alt=ver1;
        Print[dege," = ",minvall," * "," Sıralama : ",alt];
        minval=minvall;
        RatInc=True
      ]
    ]],!RatInc];
Print["ENİYİ ÇÖZÜM = ",alt(*./texttonum*)];
Print[alt./texttonum]
]

```