Paralel Tezgahlarda Yükleme ve Çizelgeleme Problemi İçin Karma Tamsayılı Modelleme ve Genetik Algoritma Temelli Yeni Bir Çözüm Yaklaşımı

Esra Erbaşta

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Şubat-2010

Mixed Integer Modeling for Parallel Machine Loading and Scheduling Problem and a New Genetic Algorithm Based Solution Approach

Esra Erbaşta

MASTER OF SCIENCE THESIS

Department of Industrial Engineering

February-2010

Paralel Tezgahlarda Yükleme ve Çizelgeleme Problemi İçin Karma Tamsayılı Modelleme ve Genetik Algoritma Temelli Yeni Bir Çözüm Yaklaşımı

Esra Erbaşta

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Endüstri Mühendisliği Anabilim Dalı
Yöneylem Araştırması Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Muzaffer Kapanoğlu

Şubat-2010

ONAY

Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Esra Erbaşta'nın YÜKSEK LİSANS tezi olarak hazırladığı "Paralel Tezgahlarda Yükleme ve Çizelgeleme Problemi İçin Karma Tamsayılı Modelleme ve Genetik Algoritma Temelli Yeni Bir Çözüm Yaklaşımı" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman **:** Doç. Dr. Muzaffer Kapanoğlu

Yüksek Lisans Tez Savunma Jürisi:

Üye : Doç. Dr. Muzaffer Kapanoğlu

Üye : Prof. Dr. Emin Kahya

Üye : Doç. Dr. Mujgan Sağır Özdemir

Üye : Doç. Dr. Şenol Erdoğmuş

Üye : Yrd. Doç. Dr. Aykut Arapoğlu

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Üretim esnasında biriken işlerin paralel tezgahların boşalması ile izin verilen sınırlı bir sürede hangi tezgahta hangi sırada işleneceğinin belirlenmesi problemi yarı-dinamik paralel tezgah çizelgeleme problemi olarak tanımlanabilir. Bu çalışmada paralel fakat tümüyle aynı olmayan tezgahların yükleme ve üretim partilerini sıralama problemi ele alınmıştır. Problemin çözümüne yönelik, hem matematiksel model, hem de genetik algoritma geliştirilip, gerçek-hayat test problemleri kullanılarak elde edilen sonuçlar karşılaştırılmış ve çözüm yöntemlerinin performansları ortaya konmuştur. Geliştirilen matematiksel modelin çözümleri CPLEX ve Lingo ortamlarında elde edilmiştir. Önerilen genetik algoritma ise Excel VBA ortamında programlanmıştır. Her iki yaklaşımın çözüm etkinlikleri sabit bir çalışma süresi için deneylemeye tabi tutulmuştur. Küçük çaplı problemlerde eniyileme yöntemlerinin ve genetik algoritmanın aynı sonuçlara erişmelerine rağmen, orta boyutlu problemlerde eniyileme yaklaşımları mevcut çözücülerin yakınsadığı alt eniyi çözümler genetik algoritmanın o süre içinde bulduğu çözümlerden geride kalmıştır. Büyük boyutlu problemlerde ise verilen süre içinde eniyileme yöntemleri herhangi bir uygun çözüm bulamaz iken geliştirilen genetik algoritma oldukça olumlu sonuçlar elde etmiştir.

Anahtar Kelimeler: Paralel tezgah çizelgeleme problemi, çoklu gezgin satıcı problemi, genetik algoritma, kablo kesme ve krimpleme problemi

SUMMARY

Semi-parallel machine scheduling is a loading and seguencing problem which decides that jobs accumulated in a job pool during the production process will be processed at which machine and in which sequence. In this work, parallel but not necessarily identical machines loading and sequencing problem is considered. Both a mathematical model and a genetic algorithm has been developed, and real-world test problems have been used to compare the solutions obtained and the performances of the solution methods have been analyzed. The developed matematical model has been solved by using GAMS-Cplex and LINGO 6.0. Proposed genetic algorithm has been programmed on the Excel VBA platform. Efficiency of both solution approach has been tested for a fixed computation time. Although the optimization method and the genetic algorithm have reached the same solution in the small scale problem, in the medium scale problems, sub-optimal solutions provided by optimization tools have been outperformed by the developed genetic algorithm. In the large scale problem, while the optimzation tools can not reach any feasible solution, developed genetic algorithm provided rather satisfactory solutions.

Key words: Parallel machine scheduling problem, multiple traveling salesman problem (MTSP), genetic algorithm, wire cutting and crimping problem

TEŞEKKÜR

Bu çalışmada, bana danışmanlık ederek, beni yönlendiren ve destekleyen, her türlü olanağı sağlayan danışman hocam Doç. Dr. Muzaffer Kapanoğlu'na teşekkürü bir borç bilirim.

ESRA ERBAŞTA

İÇİNDEKİLER

sayfa

ÖZET ..... **v**

SUMMARY ..... **vi**

TEŞEKKÜR ..... **vii**

ŞEKİLLER DİZİNİ ..... **vii**

ÇİZELGELER DİZİNİ ..... **xi**

1. GİRİŞ **1**

2. PARALEL TEZGAH ÇİZELGELEME PROBLEMİ **3**

2.1. Paralel Tezgahların Yüklenmesi ve Sıralanması Problemleri 3

2.1.1 Statik Paralel Tezgah Çizelgeleme 3

2.1.2 Dinamik Paralel Tezgah Çizelgeleme 6

2.2. Çoklu Gezgin Satıcı Problemi 7

2.2.1. Atama tabanlı tamsayılı programlama modeli 13

2.2.2. Laporte ve Nobert'in modeli 15

2.2.3. Akış tabanlı modelleme 16

3. KABLO KESME VE KRİMPLEME PROBLEMİ **18**

3.1. Sistemin Tanıtımı 18

3.2. Sistemin iş akışı 20

4. ÖNERİLEN ÇÖZÜM YAKLAŞIMLARI **22**

4.1.Tezgah Yükleme ve Sıralama Probleminin Matematiksel Modeli 22

4.2. Geliştirilen Açgözlü Genetik Algoritma 23

4.2.1 Genetik algoritmaların genel tanıtımı 24

4.2.2. Geliştirilen genetik algoritma 29

4.2.2.1. Kromozom yapısı ve ilk neslin türetilmesi 29

4.2.2.2. Mutasyon 32

4.2.2.3. Çaprazlama 34

4.2.2.4. Seçim yöntemi 35

4.3. Matematiksel Model İçin Kullanılan Çözücüler 36

4.4. Problem Seti 38

4.5. Çözüm Sonuçları 39

SONUÇ VE ÖNERİLER..... 42
KAYNAKLAR DİZİNİ..... 43
EKLER 47

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3. 1. Tek üretim hücresinin yerleşimi 18
Şekil 3. 2. Tek üretim hücresinin yerleşimi 18
Şekil 3. 3. Bitmiş ürün görünümü 19
Şekil 4. 1. Çaprazlama operatörleri 26
Şekil 4. 2. Kısmi eşleştirme çaprazlaması 27
Şekil 4. 3. Konum tabanlı çaprazlama 27
Şekil 4. 4. Döngü çaprazlama 28
Şekil 4. 5. Döngü çaprazlama 28
Şekil 4. 6. Bir populasyona ait kromozom gösterimi 30
Şekil 4. 7. Bir populasyona ait iş kromozom gösterimi 30
Şekil 4. 8. Bir populasyona ait tezgah kromozomu gösterimi 30
Şekil 4. 9. Tezgah ve iş kromozomlarının birlikte vei taşıdıkların anlamların gösterimi 30
Şekil 4. 10. Geliştirilen genetik algoritmada ilk nesil oluşturma adımları 32
Şekil 4. 11. I. Tip mutasyon gösterimi 33
Şekil 4. 12. II.Tip mutasyon gösterimi 34
Şekil 4. 13. Çaprazlama öncesi ebeveyn kromozomlar 34
Şekil 4. 14. Çaprazlama işlemi 35
Şekil 4. 15. Çaprazlama sonrası çocuk kromozom 35
Şekil 4. 16. Çaprazlama işlemi adımları 35
Şekil 4. 17. Problem H için geliştirilen genetik algoritmanın 10 dakikalık zaman diliminde aldığı değerler 41

ÇİZELGELER DİZİNİ

Sayfa

Tablo 3. 1. Ürün çeşitleri ve işlemleri 19
Tablo 4. 1. Problem boyutları 39
Tablo 4. 2. Genetik algoritmada kullanılan parametreler 39
Tablo 4. 3. Farklı çözüm yöntemleri ile ulaşılan sonuçlar..... 41

BÖLÜM 1

GİRİŞ

Üretim esnasında biriken işlerin paralel tezgahların boşalması ile izin verilen sınırlı bir sürede hangi tezgahta hangi sırada işleneceğinin belirlenmesi problemi yarı-dinamik paralel tezgah çizelgeleme problemi olarak tanımlanabilir. Bu çalışmada paralel fakat tümüyle aynı olmayan tezgahların yükleme ve üretim partilerini sıralama problemi ele alınmıştır. Paralel tezgah çizelgeleme problemleri klasik çizelgeleme problemlerinden sadece sıralama değil, yanısıra tezgahlara işlerin atanması problemini de içerdiğinden farklılık gösteririr. Bu açıdan literatürde zor problemler arasında yer alan sıralama (sequencing) ve kutu paketleme (bin-packing) problemlerini birarada çözmeyi gerektirir. Bu tarz problemlerin çözüm yöntemleri arasında birçok farklı metod bulunmasına karşın bu çalışmada eniyileme ve meta-sezgisel yaklaşımları üzerinde durulmuştur.

Tamsayılı matematiksel modelleme, kesikli üretim sistemlerinin tamamında son derece kullanışlı modelleme olanakları getirmesine karşın, en iyi çözüme götürecek çözüm algoritmalarının eksikliği yüzünden kullanımı oldukça sınırlıdır. Tamsayılı modelin küçük olmayan hemen hemen tüm durumları için en iyi çözümü garanti etmek çok fazla işlem yükü ve bilgisayar zamanı gerektirmektedir. Bu amaçla dal-sınır tekniği yaygın şekilde kullanılmaktadır. Diğer bilinen yöntemlerin pek çoğu özel problemler için belirli koşul ve kısıtlar altında dahi etkin algoritmalar olmaktan uzaktır. Dolayısıyla tamsayılı modellemenin getirdiği modelleme avantaj ve esnekliği, çözüm aşamasında çekiciliğini yitirmektedir. Bu nedenle, en iyiye yakın çözümler veren sezgisel yöntemler kısa zamanda çözüm elde edebilmesinden dolayı tercih edilebilirler. Çoklu gezgin satıcı problemlerinde ise NP-tam sınıfına dahil olması nedeniyle, problemin çözümünde sezgisel yaklaşımın kullanılması yerinde bir karar olacaktır. Meta-sezgisel bir yöntem olan genetik algoritmalar ise bu tarz çözüm yaklaşımları içinde öne çıkan rassal arama tabanlı bir yöntemdir. Bu yöntemin temelinde, biyolojideki evrim sürecinin taklit edilmesi yatmaktadır. Genellikle, bilinen eniyileme yöntemleri ile çözülemeyen ya da çözüm zamanı problemin büyüklüğü ile üstel artan problemlerde kullanılan GA, en iyi ya da en iyiye yakın çözümler vermektedir.

Bu çalışmanın ikinci bölümünde paralel tezgah çizelgeleme problemine yönelik literatüre yer verilmiş ve çoklu gezgin satıcı problemi üzerinde durulmuştur. Üçüncü bölümde kablo kesme ve krimpleme problemi açıklanmış ve sistemin iş akışına yer verilmiştir. Dördüncü bölümde çözüm yaklaşımları incelenmiş ve geliştirilen genetik algoritma ve matematiksel model açıklanmıştır. Ardından önerilen çözüm yöntemlerinin farklı gerçek-yaşam problemleri üzerinde performansları incelenmiştir.

BÖLÜM 2

PARALEL TEZGAH ÇİZELGELEME PROBLEMİ

Çoğu işlemin mevcut n tezgahtan herhangi birinde yapılabileceği durumlarda hangi işlerin hangi tezgahlarda ve hangi sıralarda yapılması halinde etkinlik ölçütünün eniyileneceğini inceleyen problemlere paralel tezgah çizelgeleme problemi denir. Klasik çizelgeleme problemlerinden sadece sıralama değil, yanısıra tezgahlara işlerin atanması problemini de içerdiğinden farklılık gösteririr. Bu açıdan literatürde zor problemler arasında yer alan sıralama (sequencing) ve kutu paketleme (bin-packing) problemlerini birarada çözmeyi gerektirir. İzleyen kesimde bu konudaki çalışmalara yer verilmiştir.

2.1. Paralel Tezgahların Yüklenmesi ve Sıralanması Problemleri

Literatürde, sıralamaya bağlı hazırlık zamanlarını göz önünde bulunduran paralel tezgah çizelgelemesi problemi üzerine yapılmış pek çok çalışma mevcuttur. Bu çalışmaları statik ve dinamik paralel tezgah çizelgeleme problemleri olarak iki başlıkta gruplamak olanaklıdır.

2.1.1 Statik Paralel Tezgah Çizelgeleme

Statik paralel tezgah çizelgeleme problemleri işlerin çizelgeleme anında sistemde aynı anda hazır olduğu varsayımına dayanır. Aşağıda bu problemleri ele alan yakın dönem çalışmaları sunulmuştur.

Zhi-Long ve Warren, m adet paralel tezgah ve n adet işin çizelgelenmesi problemini tamsayılı matematiksel model olarak yapılandırıp, problemin çözümünde Dantzig-Wolfe ayrıştırma yöntemini kullanmışlardır. Bu ayrıştırma yönteminin temelinde, dal-sınır çözüm algoritması yatmakta olup, her düğüm her alt problemin doğrusal yakınlaştırma problemidir. Herbiri bir tek tezgahtaki çizelgelemeyi temsil eden ve tek tezgah alt probleminin çözümüyle elde edilen bu doğrusal yakınlaştırma problemi sütun türetme (column generation) yaklaşımı ile çözülür. Yönetilmesi çok

daha kolay olan ayrıştırılmış modeldeki değişkenlere uygulama yerine, dal-sınır algoritması ana problemin değişkenlerine uygulanır. Çalışmalarında, toplam ağırlıklı bitiş zamanını ve ağırlıklı gecikmiş iş sayısını enküçükleyen iki farklı amacı birlikte ele almışlardır. Hesaplamasal sonuçlar, ayrıştırma yaklaşımının büyük ölçekli problemlerin çözümünde daha yetenekli olduğunu göstermiştir (Zhi-Long and Warren, 1996).

Mateus vd, bağımsız paralel tezgahların hepsinde sadece bir kez işlenebilen ve herhangi bir önceliğe sahip olmayan işlerin çizelgelenmesi problemi üzerinde durmuşlardır. Her iş, teslim zamanı, ağırlık ve her tezgah için ilgili işlem süresi ve sıralamaya bağlı hazırlık zamanlarına sahiptir. Amaç fonksiyonu, işlerin toplam ağırlıklı gecikmelerinin enküçüklenmesi olarak belirlenmiştir. Bu çalışma, problemin zaman indeksli lagrange yaklaşımı tabanlı gecikmemiş gevşet-ve-buda algoritmasını (non-delayed relax-and-cut algorithm) önermiştir. Bu yaklaşımın kullanılmasıyla, 180 iş ve 6 tezgahlık bir çizelgeleme probleminin makul bir sürede eniyi sonuca ulaşma ihtimali sözkonusu olabilmektedir (Mateus Rochade Paula, et al., 2010).

Naderi vd, melez esnek iş akışı olarak adlandırılan, iş akış çizelgelemesinin gerçekçi bir versiyonu üzerinde durmuşlardır. Bu yaklaşım, sürekli iş akışının ve paralel tezgah problemlerinin özelliklerini harmanlamıştır (her aşamada tek bir tezgah yerine paralel tezgahların bulunduğu aşamaları gözönünde bulundurmuştur). Ayrıca tüm işlerin tüm aşamalara uğrama zorunuluğunun olmadığı aşama atlamanın gerçekleşebildiği esnek versiyonu araştırmışlardır. Ayrıca her aşamada iş sırasına bağlı hazırlık zamanları bulunan işleri de gözönüne almışlardır. Esnek ve problemin hazırlık karakterisitiğine uygun iki adet ileri düzey algoritma önermişlerdir. İlk algoritma dinamik sevk kuralı sezgiseli, ikincisi ise ardıştırmalı yerel arama meta-sezgiselidir. Önerilen algoritmanın performansnı yedi farklı algoritma ile karşılaştırılmıştır (Naderi B., et al., 2010).

Huang vd ise sıralamaya bağlı hazırlık zamanı olan ve hazırlık işlemlerinin tek bir sunucudan sağlandığı paralel tezgahlardaki çizelgeleme problemi üzerinde durmuşlardır. Sistemdeki en uzun tamamlanma zamanını enküçüklemeyi amaçlayan bir tamsayılı matematiksel model geliştirmişler ve bunun alt sınırını yapılandırılmışlardır. Problemin özel bir durumunun çözümüne polinom dağılan bir sürede ulaşılmışlar, genel durumlar için ise melez bir genetik algoritma geliştirilmişlerdir. Algoritma baskı

endüstrisinde, hem rassal olarak türetilen veri setleri hem de gerçek-hayat veri setleri üzerinde test edilmiştir. Hesaplamasal sonuçlar, her iki tip veri setlerinde de yöntemin etkin ve etkili olduğunu göstermiştir. (Huang , et al., 2010)

Fröhlich vd, özdeş olmayan paralel tezgahlardaki işlerin gruplandırılması ve çizelgelenmesi problemi için iki modelleme yaklaşımı sunmuşlardır. Her iş arasında hazırlık zamanı bazen gerekli olmasına karşın iş sıralamasına bağlı olarak bazen de hazırlık süresine ihtiyaç duyulmamaktadır. Tezgah hazırlık işlemleri tek bir çalışan tarafından yapılmaktadır. Bu çalışan aynı zamanda sadece bir tek tezgahın hazırlık işlemini gerçekleştirebildiği için, bu çalışanın da çizelgelenmesi gerekmektedir. Buna göre, teslim zamanlarını karşılayan ve tüm tezgahlardaki iş yükünü enbüyükleyen bir model geliştirmişlerdir. Birinci yaklaşımda karma tamsayılı doğrusal modeli kullanırken, ikinci yaklaşımda renklendirilmiş Petri Nets tabanlı bir benzetimi kullanmışlardır. (Fröhlich R, et al., 2009)

Behnamian vd, sıralamaya bağlı hazırlık zamanına sahip paralel tezgah çizelgelemesinde enbüyük tamamlanma süresini enküçükleyecek bir melez meta-sezgisel yöntem önermişlerdir. Çözüm yaklaşımı güçlü, hızlı ve basit yapılı olup üç ana elemandan oluşmuştur. Buna göre başlangıç populasyonunu karınca kolonisi eniyilemesi (ACO) yöntemi temel alarak oluşturmuşlar, çözüm evrimi için tavlama benzetimi (SA) ve populasyonun iyileştirilmesi için değişken komşuluk araması (VNS) yöntemlerine başvurulmuşlardır. Bu üç yöntemin herbirinin avantajlarını birleştiren melezleme aşaması bu yaklaşımın en belirgin yeniliğini oluşturmaktadır. (Behnamian J., et al.,2009)

Jing Qian vd birçok tezgahta çizelgelenmiş işlerin ayrılabilir olduğunu gözönüne alan hazırlık zamanlarının enküçüklenmesini amaçlayan bir problem üzerinde durmuşlardır. Çalışmalarında öncelikle, başlangıç uygun çözümü elde etmek için sezgiselden faydanılmış ve bu başlangıç çözümü geliştirmek için komşuluk arama yaklaşımı uygulamışlardr. (Jing Qian, et al., 2009)

Xiang ve Lee, çeşitli ürünler, süreçler ve bozuklukları ile gerçek-hayat imalat sistemleri için etkili bir ajan tabanlı dinamik çizelgeleme yöntemi üzerinde durmuşlardır. Değişen koşullara uyarlanabilen özerk bir ajan yapmak ve verimli

küresel performans artışı sağlamak için karınca koloni zekası (ACI) ve yerel ajan koordinasyonunun kombine edilmesi önerilmiştir. Bu çalışma diğer dinamik çizelgelemelerden iki alanda farklılık göstermektedir. Birincisi, sıra bağımlı hazırlık kısıtları, değişken bozuklukları, çoklu ürün çeşitleri, çoklu-paralel ve çok amaçlı tezgahları ile daha genel ve gerçekçi bir üretim modeli olmasıdır. İkincisi ise, sadece görev tahsisi sorununu değil aynı zamanda görev sıralama problemini de çözen tezgah acentası ve iş acentasını kombine eden ACI dır. (Xiang and Lee, 2008)

Sıralamaya bağlı hazırlık zamanları ve tezgah-iş yeterliliği dikkate alınarak , dinamik ilişkisiz paralel tezgah çizelgelemesi üzerinde durulan bir başka çalışmada ise, Q-öğrenme algoritması çözüm yöntemi olarak uygulanmıştır. Q-öğrenme algoritmasını uygulamak için, yarı-markov karar sürecini (SMDP) yapılandırarak çizelgeleme problemini güçlendirme öğrenme problemine çevrilmiştir. Q-öğrenme algoritmasından elde edilen sonuçla beş farklı sezgiselle elde edilen sonuçları karşılaştırılmış ve sonuçta Q-öğrenme algoritmasının bu beş sezgisele göre çok daha iyi sonuçlar verdiğini tespit etti. (Zhicong Zhang, et al., 2007)

2.1.2 Dinamik Paralel Tezgah Çizelgeleme

Dinamik paralel tezgah çizelgeleme problemleri işlerin sisteme değişik zamanlarda erişebilmesine izin veren özel bir paralel tezgah çizelgeleme problemidir. Bu kapsamda yapılmış çalışmalar statik paralel tezgah çizelgelemeye göre çok sınırlı olup bu kesimde kısaca tanıtılmışlardır.

Zne-Jung Lee , Shih-Wei Lin ve Kuo-Ching Ying, sıralamaya bağlı hazırlık zamanı gerektiren paralel tezgah çizelgeleme problemi üzerinde durmuşlar ve çözüme tavlama benzetimi (restricted simulated annealing (RSA) algorithm) algoritmasıyla ulaşmışlardır. Çalışmalarında, en iyi komşuluk çizelgelemesini bulmak için sınırlı arama stratejisi ile etkili olmayan iş hareketlerinin elenmesini birleştiren sınırlı tavlama benzetimi algoritması sunmuşlardır. Önerdikleri RSA algoritması dinamik paralel tezgah çizelgelemesindeki enbüyük gecikmeyi enküçüklerken arama işlemlerini önemli ölçüde azaltmıştır. Genişletilmiş hesaplamasal denemeler, önerdikleri RSA algoritmasının basit tavlama benzetimine ve mevcut diğer algoritmalara göre çok daha fazla etkili olduğunu göstermiştir. (Zne-Jung Lee, et al., 2010)

Ying vd ise iterasyonlu açgözlü sezgisel yaklaşım önermişlerdir. Bu yaklaşımın, bu tarz problemlerin çözümünde kullanılan diğer güncel algoritmalara kıyasla çok daha etkili olduğu belirlenmiştir. (Ying Kuo-Ching, et al., 2010)

Yarı iletken silikon devre levhası üretim tesisindeki yayılma ve oksitleme alanlarında yer alan paralel tezgahların çizelgeleme problemi Li vd tarafından incelenmiştir. Farklı iş ailelerine sahip olan paralel tezgahlardaki toplam ağırlıklı gecikmeyi enküçüklemek amaçlanmıştır. Dinamik iş gelişi ve sıralamaya bağlı hazırlık zamanı ile ileri düzey proses kontrolüne yönelik kalite işlemi gereksinimleri de sistemde mevcuttur. Problemin NP-zor olması nedeniyle, makul bir sürede tatmin edici bir çözüm elde etmek için çözümde karınca kolonisi eniyileme algoritması kullanılmıştır. Önerilen yöntemin etkinliğini göstermek için geniş benzetimsel deneyler üzerinde çalışmışlardır. (L. Li , et al., 2009)

Bu çalışmada ele alınan problem yarı-dinamik bir paralel tezgah çizelgeleme problemidir. Çünkü montaj hatlarında ürünlerin hangi sırada üretildiğine ve ilgili kablonun sistemde var olup olmamasına bağlı olarak kablo hazırlık bölümüne kanban yoluyla açılan kablo siparişlerinin ürün çeşitlilik dağılımının önceden bilinmesi olanaksızdır. Buna karşın, açılan kablo siparişlerinin sipariş havuzunda 2 saatlik üretim süresince biriken işlerin verilen 10 dakikalık süre içinde toplu olarak çizelgelenmesi istenmektedir. Bu iki özelliğin bir arada bulunması probleme yarı-dinamik bir özellik katmaktadır.

2.2. Çoklu Gezgin Satıcı Problemi

Gezgin satıcı problemleri, bir çok birleşimsel en iyileme probleminin temelini oluşturan üzerinde en çok araştırma yapılmış problem tiplerinden biridir. Basit bir yapıya sahip olmasına karşın, NP-tam problemler arasında yer almakta ve bu yüzden GSP'lerinin çözümüne yönelik etkin bir algoritma mevcut değildir. Diğer bir deyişle GSP'nin çözüm süresi şehir sayısıyla üstel şekilde artmaktadır. Örneğin sadece 100 adet şehrin çizelgelenmesinde bile çözüm için çok fazla CPU zamanı gereklidir ($3*10^{141}$). Amaç, belirli sayıdaki şehir arasındaki en kısa mesafeli rotayı bulmaktır. Gezgin satıcı problemleri planlama, lojistik gibi birçok alanda uygulanmakta olup buralardaki sınırlı kaynaklar ve zaman kısıtları, problemi daha zor bir hale getirmektedir. GSP'lerinin çıkış noktası net olmamakla beraber 1832 yılında çıkarılmış

gezgin satıcı el kitabında probleme değinilmiş olup, Almanya ile İsviçre arasındaki örnek bir rotaya yer verilmiştir. Ancak herhangi bir modellemeye değinilmemiştir. GSP ile ilgili matematiksel problemler 1800'lü yıllarda İrlandalı matematikçi W.R. Hamilton ve İngiliz matematikçi Thomas Kirkman tarafından ele alınmıştır. GSP'nin genel formu hakkında ilk kez 1930'lu yıllarda Viyana'da ve Harvard'ta Karl Menger tarafından araştırma yapılmıştır. Karl Menger problemi tanımlamış, brute force algoritmasını önermiş ve en yakın komşu sezgiselinin en iyi olmadığını gözlemlemiştir. 1950 ler ve 1960 larda, problem Avrupa ve Amerikada artan şekilde popüler olmuştur. George Dantzig, Delbert Ray Fulkerson ve Selmer M. Johnson problemi tamsayılı doğrusal model olarak açıklamış ve kesme düzlemi metodunu geliştirmişlerdir. 1970'in sonlarında ve 1980'lerde Grötschel, Padberg, Rinaldi ve diğerlerinin, kesme düzlemi ve dal-sınır algoritmaları ile 2392 şehre kadar çözüm bulmalarıyla bu konuda büyük bir gelişme kaydedilmiştir. 1990'larda, Applegate, Bixby, Chvatal ve Cook, birçok güncel kayıtlı çözümde kulanılan "Concorde" programını geliştirmişlerdir. Gerhard Reinelt, birçok araştırma grubu tarafından sonuçları karşılaştırmada kullanılan ve farklı zorluk derecelerindeki örnekler için karşılaştırma kriterlerinin bir derlemesi olan TSPLIB'i 1991'de yayınlamıştır (Applegate, 2007).

Gezgin satıcı probleminin farklı bir versiyonu olan çoklu gezgin satıcı problemi (ÇGSP), aynı istasyondan çıkıp aynı istasyona dönen m adet gezgin satıcının n adet şehri bir defa gezmesi kaydıyla toplam seyahat uzunluğunun en küçüklenmesini sağlayacak turların bulunması problemidir. Buna göre, problemin (n+m-2)! olası çözümü mevcuttur (Dennis Francis Roerty, 1974). Buna göre;

- m adet gezgin satıcı vardır.
- Gezgin satıcılar aynı istasyondan ayrılırlar ve turları bittikten sonra yine aynı istasyona geri dönerler.
- Tüm şehirler sadece bir gezgin satıcı tarafından ancak bir kez ziyaret edilebilir.

Problemin çeşitleri

- Tekli / çoklu çıkış istasyonu: Tek çıkış istasyonu olduğu durumlarda, tüm gezgin satıcılar turlarına tek bir çıkış noktasından başlayıp, turlarını o çıkış noktasında sonlandırırlar. Çoklu çıkış istasyonu olduğu durumlarda, eğer her bir gezgin

satıcı için ayrı çıkış noktaları var ise, o gezgin satıcı turunu tamamladıktan sonra ya çıkış yaptığı istasyona geri döner ya da her biri ayrı istasyonda olacak şekilde farklı istasyonlara geri dönerler. İlk durum sabit varış istasyonlu durum, ikinci durum ise sabit olmayan varış istasyonlu durum olarak adlandırılır.

- Gezici satıcı sayısı: Sabit veya sınırlı bir değişken olabilir.
- Sabit maliyetler: Gezgin satıcı sayısı problemde sabit değilse, çözümde kullanılan her gezgin satıcı için sabit bir maliyet söz konusudur. Bu durumda çözümde yer alan gezgin satıcı sayısının en küçüklenmesi amaçlanabilir.
- Zaman çizelgesi: Bazı durumlarda belli şehirler (noktalar) belli bir zaman süresinde ziyaret edilme zorunluluğu vardır. Bu zorunluluk ÇGSP'nin önemli bir uzantısı olan zaman çizelgeli çoklu gezgin satıcı problemini ortaya çıkarmaktadır. Bu tarz problemlere, okul otobüsü, gemi ve uçak çizelgeleme problemleri örnek olarak gösterilebilir.
- Diğer özel sınırlamalar: Bu sınırlamalar her gezgin satıcının ziyaret etmesi gereken şehir sayısındaki ya da her bir gezgin satıcının katettiği en büyük/en küçük mesafe büyüklüğündeki sınırlamalar olarak ortaya çıkabilir.

ÇGSP, araç rotalama probleminin (VRP) bir kapasite kısıtının kaldırılmış şekli olarak düşünülebilmektedir. Bu ise , gezgin satıcılara (araçlar) uygun bir büyüklükte kapasite atanması koşuluyla araç rotalama problemi için önerilen tüm formülasyon ve çözüm yaklaşımları ÇGSP için de geçerli olabildiğini göstermektedir. Aynı şekilde gezgin satıcı sayısının tek olması durumda ise ÇGSP gezgin satıcı problemine dönüşmektedir.

Weimin, Sujian, Aiyun, Fanggeng, tütün dağıtım problemini ÇGSP olarak incelemiştir. Buna göre problemi statik olan teslimat rota programlama ve günlük dinamik araç rotalama olmak üzere iki aşamada çözmüştürler. Statik aşama, işyükü dengelemesi de yapılarak ÇGSP olarak modellenmiştir. Melez karınca kolonisi eniyileme algoritması (HACO) bu ÇGSP için geliştirilmiştir (Weimin et al., 2009). Svestka ve Huckfeldt ise yatırılan paranın bankalar arasında nakil edilmesine yönelik bir uygulama yapmışlardır. Buna göre, yatırılmış para, taşıma görevlisi tarafından bir banka şubesinden alınıp merkez bankaya götürülür. Problem taşıma görevlisinin enaz maliyetli taşıma rotasına karar vermektir (Svestka, et al., 1973). Buna benzer iki

uygulama da Lenstra ve Rinnooy Kan tarafından tanımlamışlardır. Buna göre ilk uygulama, kuzey Hollanda'daki telefon kulübelerini ziyaret etmesi gereken teknik görevlinin izlemesi gereken en az maliyetli rotayı bulmaya yöneliktir. İkinci uygulama ise en az araçla Utrecht'teki 200 adet posta kutusunu gezecek bir rota tasarlamaktır. (Lenstra and Rinnooy Kan, 1975)

Otobüs çizelgeleme problemi ise Angel tarafından bazı ek kısıtlarla ÇGSP'nin başka bir çeşidi olarak incelenmiştir. Çizelgelemenin amacı rota sayısını, tüm otobüslerin kat ettiği yol mesafesini en küçükleyen, fazla yüklemenin olmadığı ve gidiş dönüş için gerekli toplam sürenin kanunen izin verilen süreyi aşmamasını sağlayan otobüs güzergahını bulmaktır (Angel, 1972).

Gilbert ve Hofstra, ÇGSP'nin çoklu zaman aralıklı uygulamasını tanımlamışlardır. Bu uygulama, turizm endüstrisindeki tur acentası ve bayiler arasındaki görüşmelerin çizelgelenmesinden ortaya çıkmış bir problemdir. Buna göre gezgin satıcıya karşılık gelen her tur acentası, şehirleri temsil eden belirlenmiş bayi gişelerini ziyaret etmelidir (Gilbert and Hofstra, 1992).

Askeri keşif, depo otomasyonu, posta ofisi otomasyonu, gezegen araştırması, deniz tabanı incelemesi, maden önlemleri, haritalama, kurtarma gibi alanlarda kullanılan çoklu robot sistemlerinde yer alan bağımsız seyyar robotların çizelgelenmesi probleminden ortaya çıkmıştır. Belirli bir alandaki farklı hedef noktalarının ziyaret edilmesi bu problemlerdeki tipik görevleridir. Görev planlama her robotun mümkün olan en kısa zamanda görevlerini tamamlamalarını sağlayacak en iyi rotaya karar vermeye yöneliktir. Görev planlayıcı, n adet robot ve bazı robotlar tarafından ziyaret edilecek m adet amaç ve tüm robotların sonunda döneceği bir ana şehirden oluşan ÇGSP'nin bir çeşidini kullanır. Görev planlama konusundaki ÇGSP uygulaması Brummit ve Stentz tarafından ele alınmıştır. Bağımsız robotların planlaması ÇGSP'nin bir çeşidi olarak Yu tarafından modellenmiştir (Yu, et al., 1798). Benzer şekilde, insansız hava araçlarının planlamasına yönelik rotalama problemi Ryan tarafından zaman tabanlı ÇGSP olarak modellenmiştir (Ryan, et al., 1998). Çoklu robotların çoklu hedefleri ziyaret etmesine yönelik dinamik görev tahsisi algoritması ise Zhenzhen vd tarafından tanıtılmıştır (Zhenzhen, etal., 2009). Ele aldıkları problem, gezgin satıcı problemine göre modellenemediği için karınca koloni sistemi ile çözülemeyen

problemin çoklu gezgin satıcı problemi olarak modellenip, değiştirilmiş karınca koloni sistemiyle çözümüne ulaşılmıştır. Bu konuda yapılmış diğer bir araştırma ise Clark vd tarafından yapılmıştır. Bu çalışmada, diğer çoklu robot çizelgeleme problemlerinden farklı olarak robotların birbirlerine yardım edebilmesi konusu incelenmiştir. Bu sırada tek başına çalışması gereken robotların da bu özelliği modele yansıtılmıştır. Bu sayede robotların ortaklaşa çalışabildiği durumlarda takım çalışmasının arttırılması sağlanarak ortaklaşa çalışan robotların maliyetleri azaltılırken, bu robotların, tek başına çalışması gereken robotlarla birlikte çalışması engellenmiş olmuştur (Clark, et al., 2009).

Demir ve çelik endüstrisinde, bir üretim vardiyası için sıcak haddeleme sırasında oluşan toplam geçiş maliyetlerinin en küçüklenmesini sağlayacak üretim sıralamasının tasarlanması da bir ÇGSP'dir. Çin'deki demir ve çelik endüstrisinde böyle bir problemin güncel modelleme uygulaması Chen vd tarafından araştırılmıştır (Chen , et al., 2008). Burada, siparişler şehirlere karşılık gelirken, iki sipariş arasındaki geçiş maliyeti ise iki şehir arasındaki uzaklıklara karşılık gelmektedir.

ÇGSP'nin bir diğer güncel ve ilginç uygulaması, küresel navigasyon uydu sisteminin tasarımında ortaya çıkmıştır. Bu sistem, dünya çapındaki her yeri kapsayan, felaketlere karşı erken uyarı ve bunların yönetimi gibi gerçek hayat uygulamalarında hayati önem taşıyan uzay tabanlı uydu sistemidir. Amaç, dünya üzerinde veya uzayda dünya çevresinde bilinmeyen noktaların coğrafi konumunu uydu araçları kullanarak belirlemektir. Birden fazla alıcı ya da birden fazla çalışma periyodu olduğunda, problem ÇGSP olarak modellendirilebilir.

Gezgin satıcılar arasındaki iş yükünün dengelenmesi problemi de bu konuda çalışılmış başka bir örnektir (Okonjo-Adigwe, 1988). Burada ÇGSP tabanlı modelleme ve çözüm yaklaşımı, her gezgin satıcının toplam seyahat süresi ve toplam iş yükünün alt ve üst sınırları gibi bazı ek sınırlamalarla iş yükü çizelgeleme problemini çözmede kullanılmıştır. Diğer bir örnek ise gece emniyet servisi problemidir. Bu probleme göre, görevli olduğu yerleşim bölgesinde günlük incelemesini gerçekleştiren koruma görevlilerine görev ataması gerçekleştirilmektedir. Burada aynı zamanda göz önüne alınması gereken kapasite ve zaman periyodu gibi bazı kısıtlar da söz konusudur.

ÇGSP, gemi operasyon planlamasında rıhtım vinçlerinin çizelgeleme alt probleminde de kullanılmaktadır. Bu problemi inceleyen Kim ve Park, dal-sınır algoritmasındaki kritik alt sınırı bulmada ÇGSP'nden yararlanmışlardır (Kim ve Park, 2004).

Zaman tabanlı ÇGSP, büyük ölçekli nakliye problemlerinde de kullanılabilmektedir. Bu kapsamda Wang ve Regan yerel kamyon yükleme ve teslim problemini zaman tabanlı asimetrik ÇGSP olarak modellemişlerdir. Problemin çözümü zaman tabanlı ayrıştırmalar kullanılarak ardıştırmalar yapılarak elde edilmiştir (Wang ve Regan, 2002).

ÇGSP'nin bir diğer ilginç uygulaması, çoklu nesnelerin hareketini düzenleme probleminde ortaya çıkmaktadır (Basu A., et al., 2000). Elektronik devrelerin birleştirilmesinde ve depo gibi yapılandırılmış alanlardaki gezici robotların koordinasyonunda bu tarz problemler oluşmaktadır. Bu problem karelere bölünmüş alanlarda tanımlanır. Kareler nesne ya da boşlukları içerebilir. Karelere bölümlenmiş ve boş alanlar içeren bu alanlardaki nesneler için en uygun hareketler ÇGSP ile bulunabilir.

ÇGSP'nin çözüm yöntemleri arasında birçok farklı yaklaşım yer almaktadır. Bu yaklaşımlardan biri olan eniyi yöntemleri modelleme konusunda avantaj sağlarken çözüm aşamasında özellikle büyük ölçekli problemlerde çekiciliğini yitirmektedir. Bu nedenle daha farklı çözüm yöntemlerine yönelme söz konusu olmuştur. Bu yöntemlerin arasında karınca kolonisi algoritması gösterilebilir. Weimin bu algoritmadan yola çıkarak melez bir karınca kolonisi eniyileme tekniği öne sürmüştür. (Weimin, et al., 2009). ÇGSP lerinin çözümüne yönelik kullanılan bir diğer yöntem ise meta-sezgisel yöntemlerden olan genetik algoritmadır. Bu alanda yapılmış çalışmalardan, gruplamaya dayalı bir genetik algoritma sunan Brown vd nin ele aldığı çalışma ile Alok ve Anuraq'ın çalışması örnek verilebilir (Brown E.C.,et al., 2007) (Alok S. and Anuraq B., 2009). Bunlar birbirlerinden kromozom yapısı, genetik işlemciler gibi yönlerden farklılık göstermektedir.

ÇGSP için önerilen farklı tipte birçok tamsayılı programlama modeli mevcuttur. Bu modellere yönelik bazı teknik açıklamalar aşağıdaki gibidir. ÇGSP *G=(V,A)*

grafiğinde tanımlanmıştır. Burada V n adet düğümü, A ise ayrıtları (kenarları) temsil etmektedir. A ile ilişkili olan C=(c_{ij}) maliyet(uzaklık) temsil etmektedir. Burada A kümesindeki her i ve j için $c_{ij}= c_{ji}$ olması durumunda C simetrik, diğer durumda ise asimetrik olmaktadır. Eğer V kümesindeki her i ve j için $c_{ij}+ c_{jk}\geq c_{ik}$ ise C'nin üçgen eşitsizliğine cevap verdiği söylenir.

ÇGSP için literatürde farklı tamsayılı programlama modelleri önerilmiştir. Bunlar arasında atama tabanlı modeler, ağaç tabanlı model ve 3 indis akış tabanlı modeller bulunmaktadır.

2.2.1. Atama tabanlı tamsayılı programlama modeli

ÇGSP genellikle atama tabanlı iki indisli tamsayılı doğrusal programlama ile modellenmektedir.

$$x_{ij} = \begin{cases} 1, eğer\ (i,j)\ yolu\ turda\ yer\ alıyorsa \\ 0, diğer\ durumlarda \end{cases}$$

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$s.t. \sum_{j=2}^{n} x_{1j} = m, \qquad (1)$$

$$\sum_{j=2}^{n} x_{j1} = m, \qquad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 2,...,n \qquad (3)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 2,...,n \qquad (4)$$

$$+ \text{alt tur engelleme kisitlari} \qquad (5)$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A, \qquad (6)$$

(3), (4) ve (6) genel atama kısıtları iken, (1) ve (2) kısıtları ise m adet gezgin satıcının 1 nolu düğümden çıkıp aynı düğüme geri dönmesini garantilemektedir. Kısıt (5) ise alt turları engellemek için kullanılmaktadır. Bu kısıtlara alt tur engelleme kısıtları adı verilmektedir .

ÇGSP için literaturde birçok alttur engelleme kısıtları önerilmiştir. Bunlardan ilki aslında GSP için önerilmesine karşın ÇGSP için de geçerli olan Dantzig'in önerdiği yöntemdir (Dantzig et.al., 1954). Bu kısıtlar :

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1, \quad \forall S \subseteq V/\{1\}, \quad S \neq \phi \qquad (7)$$

yada

$$\sum_{i\notin S}\sum_{j\in S} x_{ij} \ge 1, \quad \forall S \subseteq V/\{1\}, \quad S \neq \phi \qquad (8)$$

(7) ve (8) nolu kısıtlar çözüm için bağlayıcı gerekliliktir. Örneğin S'nin ana noktayı içermeyen alt turlarını engeller. Ancak bu kısıtın sayısı düğümlerin artmasıyla üstel şekilde arttığı için ne problemin çözümünde ne de bunun doğrusal programlama yaklaşımında pratik bir çözüm oluşturmaz. Miler, düğüm potansiyelleri diye adlandırılan ve alttur engelleme kısıtlarının polinomsal sayılarda kalmasını sağlayan $O(n^2)$ ilave sürekli değişkenler ekleyerek bu problemin üstesinden gelmişlerdir (Miller C.E, et al., 1960). Buna göre önerilen kısıt aşağıdaki gibidir.

$$u_i - u_j + px_{ij} \le p - 1 \quad \forall\, 2 \le i \ne j \le n \qquad (9)$$

Burada p herhangi bir gezgin satıcı tarafından ziyaret edilen en büyük düğüm sayısını temsil etmektedir. Her düğümün olası düğümleri, turdaki ilgili düğüm sırasını göstermektedir.

ÇGSP için önerilen ve yeni sütun ve satırlarla orijinal maliyet matrisinin büyümesini gerektiren başka bir alttur engelleme kısıtı ise Svestka ve Huckfeldt tarafından önerilmiştir. (Svestka ve Huckfeldt, 1973).

$$u_i - u_j + (n-m)x_{ij} \le (n-m-1), \qquad \begin{array}{l} i = m+1,\ldots n+m-1 \\ j = m+1,\ldots,n+m-1 \\ i \ne j \end{array}$$

Ancak Gavish, kısıtların m≥2 için doğru olmadığını göstermiş ve aşağıdaki kısıtı elde etmiştir (Gavish, 1976).

$$u_i - u_j + (n-m)x_{ij} \le n - m - 1 \quad \forall 2 \le i \ne j \le n \qquad (10)$$

ÇGSP için en güncel alttur engelleme kısıtı ise Kara ve Bektaş tarafından önerilmiştir. Bu kısıtlar diğer var olan kısıtlara ilave yan kısıtları da içermesiyle

farklılık göstermektedir. Bu kısıtlar bir gezgin satıcının ziyaret ettiği (K)ların alt sınırını belirlemede kullanılır. Böylesi alt sınırlar, her gezgin satıcının bazı ürünleri toplamak için en azından K adet müşteriyi gezmesi gerekliliğini doğurabilir. Bu kısıtlar:

$$u_i - (L-2)x_{1i} - x_{i1} \le L-1, \qquad i = 2,...,n, \qquad (12)$$

$$u_i + x_{1i} + (2-K)x_{i1} \ge 2, \qquad i = 2,...,n, \qquad (13)$$

$$u_i + u_j + Lx_{ij} + (L-2)x_{ji} \le L-1, \qquad 2 \le i \ne j \le n, \qquad (14)$$

Buradaki L, kısıt (11) deki L ile aynı anlamı taşımaktadır. Kısıt (12) ve (13) ise gezgin satıcının ziyaret edeceği şehir sayısının alt sınırını belirlemek için kullanılır (Kara ve Bektaş, 2004).

2.2.2. Laporte ve Nobert'in modeli

Laporte ve Nobert ÇGSP'ndeki simetrik ve asimetrik durumlar için iki farklı model göstermiştir ve her gezgin satıcı için amaç fonksiyonunda yer alan sabit bir maliyete yer vermiştir (Laporte and Nobert, 1980). Bu modelde üstel sayıda alttur engelleme kısıtı yer almaktadır. Bu modeller ayrıca önceden tanımlanmış olan iki indisli x_{ij} değişkeni tabanlıdır. Asimetrik ÇGSP için model:

$$\min \sum_{i \ne j} c_{ij}x_{ij} + fm$$

$$s.t. \sum_{j=2}^{n} x_{1j} + x_{j1} = 2m, \qquad (15)$$

$$\sum_{i \ne k} x_{ik} = 1, \qquad k = 2,...,n \qquad (16)$$

$$\sum_{j \ne k} x_{kj} = 1, \qquad k = 2,...,n \qquad (17)$$

$$\sum_{i \ne j; i,j \in S} x_{ij} \le |S| - 1 \quad 2 \le |S| \le n\text{-}2, \quad S \subseteq V/\{1\}, \qquad (18)$$

$$x_{ij} \in \{0,1\}, \forall i \ne j, \qquad (19)$$

$$m \ge 1 \text{ ve tamsayı} \qquad (20)$$

Bu model çözümde kullanılan gezgin satıcı sayısını ve seyahatın toplam maliyetini enküçükleyen saf 0-1 tamsayı modelidir. Kısıt(16) ve (17) standart atama kısıtları ve kısıt(18) Dantzig'in önerdiği alttur engelleme kısıtıdır (Dantzig, et al., 1954). Laporte ve Nobert'in simetrik ÇGSP için modeli:

$$\min \sum_{i \neq j} c_{ij} x_{ij} + fm$$

$$s.t. \sum_{j=2}^{n} x_{1j} = 2m, \qquad (21)$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad \text{k} = 2,...,n \qquad (22)$$

$$\sum_{\text{i}\neq\text{j;i,j}\in\text{S}} x_{ij} \leq |S| - 1 \quad 3 \leq |S| \leq n\text{-}2, \quad \text{S} \subseteq \text{V}/\{1\}, \qquad (23)$$

$$x_{ij} \in \{0,1\}, \quad 1 < i < j, \qquad (24)$$

$$x_{1J} \in \{0,1,2\}, \quad j = 2,..,n \qquad (25)$$

$$\text{m} \geq 1 \text{ ve tamsayı} \qquad (26)$$

Bu modeldeki ilginç olan nokta x_{1j}'nin aldığı 0,1 ve 2 değerlerinden dolayı saf bir 0-1 tamsayılı model olmamasıdır. Dikkat edilirse problemin simetrik olmasından ve çözümde kullanılan her kenarı göstermek için uygun olan sadece tek bir değişkenin olmasından dolayı, x_{ij} değişkeni sadece i<j ler için tanımlanmıştır. Kısıt (21) ve (22) ana nokta ve diğer noktalar arasındaki derece kısıtlarıdır. Diğer kısıtlar önceden tanımlandığı gibidir.

2.2.3. Akış tabanlı modelleme

Christofides üç indisli araç rotalama modellemesi önermişlerdir (Christofides, 1981). Buradaki kapasite ve maliyet kısıtlarının model dışı bırakılmasıyla, bu model ÇGSP'e uyarlanabilir. Bu model araç rotalama problemi için önerilmiş olmasına karşın burada ÇGSP'e uyarlanmış haline yer verilecektir. Modelin karar değişkeni,

$$x_{ijk} = \begin{cases} 1, eğer\ araç\ k\ j\ noktasını\ i\ noktasından\ Hemen\ sonra\ ziyaret\ ederse \\ 0, diğer\ durumlarda \end{cases}$$

Model;

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \sum_{k=1}^{m} x_{ijk}$$

$$s.t. \sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1, \qquad j = 1,..,n, \tag{35}$$

$$\sum_{i=1}^{n} x_{ipk} - \sum_{j=1}^{n} x_{pjk} = 0, \qquad k = 1,..,m, p = 1,..,n \tag{36}$$

$$\sum_{j=1}^{n} x_{1jk} = 1, \qquad k = 1,..,m, \tag{37}$$

$$u_i - u_j + n\sum_{k=1}^{m} x_{ijk} \le n-1, \qquad i \ne j = 2,..,n \tag{38}$$

$$x_{ijk} \in \{0,1\}, \forall i, j, k. \tag{39}$$

Kısıt (35) her müşterinin sadece bir kez ziyaret edilmesi gerektiğini, kısıt (36) gezgin satıcı bir müşteriyi ziyaret ettiğinde mutlaka aynı müşteriyi terkte etmesi gerektiğini belirtmektedir. Kısıt (37) her aracın sadece bir kez kullanılması gerektiğini, kısıt (38) ise Miller, Tucker ve Zemlin (MTZ) tabanlı alttur engelleme kısıtlarının üç-indisli modele göre geliştirilmiş halidir. Ancak bu modeldeki değişken sayısı, orta boyutlu bir ÇGSP'i için bile çok büyük olması nedeniyle bu modelin doğrudan çözülmesi pratik olmayacaktır.

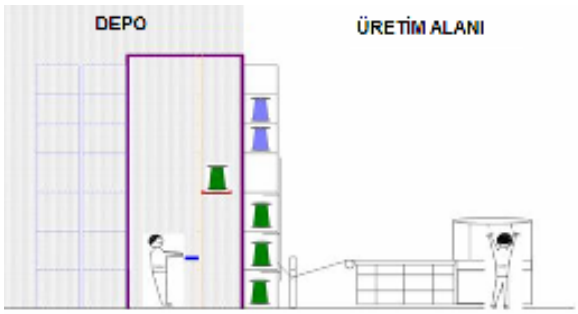
İzleyen bölümde, araştırmanın yapıldığı paralel tezgahlara sahip üretim sistemi tanıtılıp problemin yapısı incelenecektir.

BÖLÜM 3

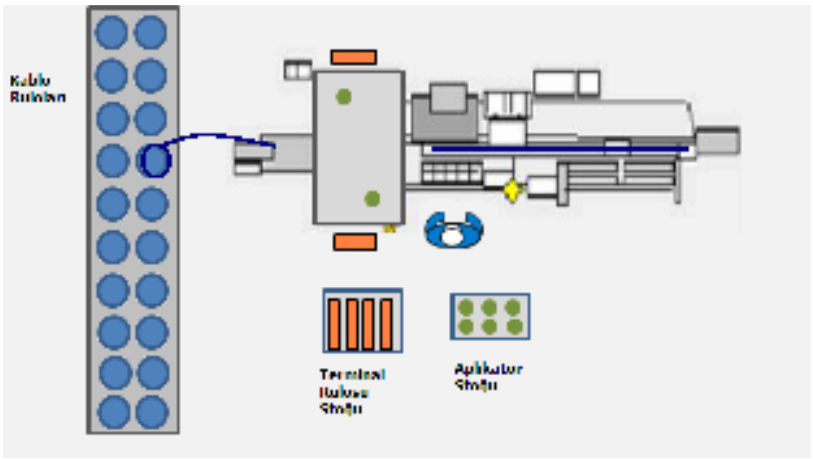
KABLO KESME VE KRİMPLEME PROBLEMİ

3.1. Sistemin Tanıtımı

Araştırmanın yapıldığı üretim sistemi, otomotiv sektörüne kablo donanımı yapan bir firmanın kablo hazırlama bölümüdür. Buna göre ele alınan sistemde 8 adet Komax-433 çift taraflı kablo kesme krimpleme ve conta takma tezgahı ve 2 adet YACC-7 çift taraflı kablo kesme ve krimpleme tezgahı mevcuttur. Buna göre tezgahlara atanan siparişler doğrultusunda tezgahlar, kabloları istenen uzunlukta keser, uç soyumunu yapıp, siparişe göre krimpleme ve conta takma işlemlerini gerçekleştirirler. Bir üretim tezgahının, malzeme ve kablo bobinlerinin üretim bölümündeki yerleşimi Şekil 3. 1'de ve Şekil 3. 2'de gösterilmiştir.



Şekil 3. 1. Tek üretim hücresinin yerleşimi



Şekil 3. 2. Tek üretim hücresinin yerleşimi

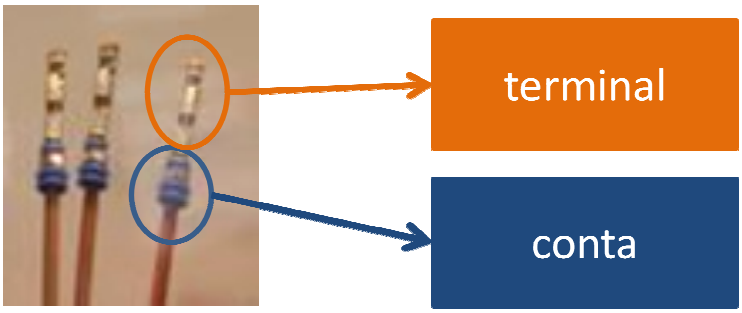
Mevcut sistemde 6 çeşit ürün üretilmektedir.

1. Her iki tarafında uç soyumunun yapılması (Tablo 3. 1.a)
2. Tek tarafa terminal basılıp diğer tarafa sadece uç soyumu yapılması (Tablo 3. 1.b)
3. Her iki tarafa da uç soyumu ve terminal basma işlemlerinin yapılması (Tablo 3. 1.c)
4. Her iki tarafa uç soyumu yapılıp tek tarafa conta takma ve terminal basma işlemlerinin yapılması (Tablo 3. 1.d)
5. Her iki tarafa da uç soyumu ver terminal basma işlemleri ve tek tarafa conta takma işleminin yapılması (Tablo 3. 1.e)
6. Her iki tarafa da uç soyumu terminal takma ve conta takma işlemlerinin yapılması (Tablo 3. 1.f)

Tablo 3. 1. Ürün çeşitleri ve işlemleri

ÖRNEKLER	A		B		C		D		E		F	
İŞLEMLER	sol	sağ	sol	sağ	sol	sağ	sol	sağ	sol	sağ	sol	sağ
Uç soyma	X	X	X	X	X	X	X	X	X	X	X	X
Terminal basma			X		X	X	X		X	X	X	X
Conta takma							X		X		X	X

Üretim aşamaları tamamlanmış, uç soyumu yapılmış, conta takılmış ve terminal basılmış bir kablonun tek ucunun görünümü Şekil 3. 3'de gösterilmiştir.



Şekil 3. 3. Bitmiş ürün görünümü

3.2. Sistemin iş akışı

Ele alınan üretim sisteminde iki çeşit kablo kesme ve krimpleme tezgahı mevcuttur. Bu tezgahlardan Komax-433 (8 adet) tezgahı kablo kesimi, uç soyumu, terminal basma ve conta takma işlemlerini yapabilirken, Yacc-7 (2 adet) tezgahı conta takma dışında tüm işlemleri yapabilmektedir. Birim kablo üretim süreleri bu iki tip tezgah içinde farklılık göstermekte olup, bu süreler EK-1 ve EK2 deki tablolarda gösterilmiştir.

Ele alınan üretim sistemi olan kablo hazırlık bölümüne yönelik siparişlerin açılması aşamaları aşağıdaki gibi gerçekleşmektedir. Müşteriden gelen ürün siparişleri üretim planlama bölümü tarafından değerlendirilip haftalık plan haline dönüştürülür. Günlük bazdaki bitmiş ürün planları kablo hazırlık bölümüne iletilir. Kablo hazırlık bölümü ihtiyaç duyulan kabloları ve miktarlarını belirledikten sonra kanban kart sayılarındaki düzenlemeleri yapar. Eğer kanban kart sayılarında geçen haftaya göre bir artış gerekiyorsa, bu kart sayısı kadar kablo hazırlık bölümü siparişi, itme tipi sipariş (push order) şeklinde kablo hazırlık bölümü tezgahlarına gönderilir. Bu ilk düzenleme aşaması bittikten sonra tüketim esnasında okutulan kanban kartları yardımıyla sistem siparişini kendi açar. Kanban kartlarının okutulmasıyla açılan siparişler sipariş havuzunda bekletilir. İki saatte bir yapılan çizelgeleme işlemiyle, havuzda bekleyen bu işlerin tezgahlara atanarak çizelgeleme işlemi yapılır. İki saat sonunda bu işlerin üretim1 bölümünde bitmiş olması gerekmektedir.

İşlerin tezgahlara atama işlemi gerçekleştirildikten sonra her tezgah başında bulunan bilgisayarlara, o tezgaha ait üretim listesi gönderilir. Buna göre operatör, sıradaki ürün için gerekli olan kablo, terminal ve aplikatör çeşitlerini kontrol eder. Bu etkenlerden biri ya da birkaçı önceki işten farklılık gösteriyorsa ilgili hammadde veya aplikatörün değişikliği gerçekleştirilir. Eğer bu etmenlerden biri değişecek olursa, üretime başlamadan önce tezgahın ayarının kontrol edilmesi için yaklaşık 10 adetlik bir test üretimi gerçekleştirilmesi gerekmektedir.

Buna göre hazırlık işlemlerini kablo değişikliği, tek/çift terminal rulosu değişikliği, tek/çift conta değişikliği ve test üretiminin yapılması işlemleri

oluşturmaktadır. Önceki işe bağlı olarak gerçekleşecek bu değişikliklere göre her ürün için gerekli olan hazırlık süreleri farklılık gösterecektir.

Farklı ürün olmasına karşın hazırlık zamanı gerektirmeyen durumlar da söz konusudur. Örneğin; aynı kablo, aplikatör ve terminalin kullanılmasına karşın çift tarafın değil de sadece tek bir tarafın terminalinin basılması diğer tarafın sadece uç soyumunun yapıldığı durumlarda test ürünü alınmamaktadır.

Hazırlık işlemlerinin bitmesinden sonra tezgah çalıştırılarak standart olarak kabul edilen lotlarda üretim yapılır. Her lotun üretimi bittikten sonra kablolar standartta belirtilen şekilde rulo haline getirilir ve terminalleri koruma amaçlı bir koruyucu başlık takılarak ürüne ait etiket takılır. Ardından bitmiş ürün önceden tanımlanmış askısına asılarak işlem bitirilir.

Bu üretim sisteminde işlerin paralel tezgahlara yüklenmesi ve üretim partilerinin sıralanması probleminde amaç toplam gecikmeleri enküçüklemektir. Hazırlık yönünden en az süreye gereksinim duyacak ürün çeşitlerinin doğru tezgahlara ve doğru sırada atamalarının gerçekleştirilmesiyle bu amaca ulaşılabilecektir. Bu yapı nedeniyle, problem çoklu gezgin satıcı problemiyle belirgin benzerlikler göstermektedir. Buna göre gezgin satıcıyı temsil eden paralel tezgahlara işlerin atanması ve bu işlerin en az toplam gecikme ile tamamlanmasını sağlayan üretim sıralarının (rotaların) belirlenmesi problemi olarak ifade edilebilinir.

BÖLÜM 4

ÖNERİLEN ÇÖZÜM YAKLAŞIMLARI

Kablo üretim sisteminde, bir kablonun üretimi farklı tezgahlarda yapılabilmektedir ve bundan dolayı da parçalar alternatif atamalara sahip olmaktadırlar. Hangi işin hangi tezgahta ve hangi sırada yapılacağının belirlenmesi için tamsayılı doğrusal karar modelleri ile çözüme ulaşma süresinin makul süreden oldukça fazla olması nedeniyle çözümde genetik algoritmadan da yararlanılacaktır.

4.1.Tezgah Yükleme ve Sıralama Probleminin Matematiksel Modeli

Ele alınan problemde parçaların seçimi rassal olarak yapılmıştır. "0" ve "120" olmak üzere işlere ait iki farklı teslim süresi mevcuttur. Bunlardan "0" teslim zamanlı işler, önceki çizelgeleme sonucu tezgahlara atanmasına karşın henüz bitirilememiş veya plan değişikliği, kanbanın zamanında okutulmaması gibi nedenlerden dolayı ürünün acile düştüğü işleri göstermektedir. "120" teslim zamanlı siparişler ise yeni çizelgelenecek ve 120 dakika sonunda bitirilmesi gereken işlerdir.

Buna göre modelde kullanılan değişken ve parametre tanımlamaları ile geliştirilen model şu şekildedir:

Karar değişkenleri:

$$y(i,j,m) = \begin{cases} 1, & \text{eğer m tezgahına i işinden hemen sonra j işi ataması yapılmış ise} \\ 0, & \text{diğer durumlarda} \end{cases}$$

k(i,m) : i işinin m tezgahındaki tamamlanma zamanının sipariş teslim zamanından pozitif sapması (i işinin m tezgahındaki gecikme süresi)

l(i,m) : i işinin m tezgahındaki tamamlanma zamanının sipariş teslim zamanından erken teslim edilme süresi

x(i,m): m. tezgahta i.işin bitis zamanı

Parametreler:

z (i,m): i işinin m tezgahındaki işlem süresi

h(i,j) : i işinden sonra j işinin yapılması durumunda gerekli olan hazırlık süresi

t(i) : i işinin teslim zamanına göre kalan süresi

s(i) : i işinin sipariş miktarı

$$\sum_{j\neq i}\sum_{m} y(i,j,m) = 1 \quad \forall i \tag{1}$$

$$\sum_{i\neq j}\sum_{m} y(i,j,m) = 1 \quad \forall j \tag{2}$$

$$\sum_{j\neq i} y(i,j,m) - \sum_{j\neq i} y(j,i,m) = 0 \quad \forall i,m \tag{3}$$

$$x(j,m) \geq x(i,m) + \left(h(i,j) + z(j,m) * s(j)\right) * y(i,j,m) + M * \left(y(i,j,m) - 1\right) \tag{4}$$
$$\forall i, j \neq i, j > m, m$$

$$x(i,m) + l(i,m) - k(i,m) = \left(t(i)\right) * \sum_{j\neq i} y(i,j,m) \qquad \forall i > m, m \tag{5}$$

$$u(i,m) - u(j,m) + (n - makine\_sayisi + 1) * y(i,j,m) <= n - makine\_sayisi \tag{6}$$
$$\forall \mathrm{i} > \mathrm{m}, \mathrm{j} > \mathrm{m}, \mathrm{m}$$

$$y_{ijm} \in \{0,1\}, \forall i \neq j, \tag{7}$$

$$\mathrm{x}_{\mathrm{ij}}, k_{im}, l_{im} \geq 0 \tag{8}$$

$$\min z = \sum_{i>1}\sum_{m} k(i,m) \tag{9}$$

Buna göre modelde kısıt (1) ve (2), her işin sadece bir tezgaha atanmasını, Kısıt(3) ise işin başladığı tezgahta bitmesini sağlamaktadır. Kısıt(4)'te, aynı tezgaha ardışık olarak atanmış işlerin bitiş zamanını belirlemeye yöneliktir. Buna göre aynı işin bitiş zamanı, o tezgahtaki bir önceki işin tamamlanma zamanı ve ilgili iş için gerekli olan hazırlık zamanı ve işlem suresinin toplamından büyük olmalıdır. Kısıt(5), işlerin teslim zamanından pozitif veya negatif sapmalarını belirlemektedir. Kısıt(6) ise, alt tur engelleme kısıtıdır. Matematiksel modelin amaç fonksiyonu ise teslim zamanından pozitif sapmaları yani gecikmeleri (tardiness) enküçüklemektir. Buna göre GAMS ve LINGO'da geliştirilen modeller EK3 ve EK4 te verilmiştir.

4.2. Geliştirilen Açgözlü Genetik Algoritma

Geliştirilen açgözlü genetik algoritma tanıtılmadan önce genetik algoritmalara ilişkin genel bilgiler izleyen kesimde sunulmuştur.

4.2.1 Genetik algoritmaların genel tanıtımı

Genetik Algoritma karmaşık çok boyutlu arama uzayında tam veya uygun çözümü bulmak için kullanılan ve bütünsel en iyiyi arama yöntemidir. Bu teknik, soyaçekim, mutasyon, doğal seçim ve çaprazlama gibi doğada gözlemlenen evrimsel süreçten esinlenilerek geliştirilmiştir.

Genetik algoritmanın temel ilkeleri ilk kez Michigan Üniversitesi'nden John Holland tarafından ortaya atılmıştır. Holland, çalışmalarını 1975 yılında "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. İlk olarak Holland evrim yasalarını genetik algoritmalar içinde eniyileme problemleri için kullanılabilmesinin kuramsal yapısını ortaya koymuştur.

Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde popülasyon adını alır. Popülasyonlar, sayı dizileri olan kromozomlardan oluşur. Bu sayı dizilerindeki her bir elemana ise gen adı verilmektedir. Kromozomların yapısı problemden probleme değişiklik göstermektedir. Genetik algoritmaların başarılı olmasındaki en önemli faktör de, problemin çözümünü temsil eden bu kromozomların oluşturulma şeklidir. Her popülasyonun amaç fonksiyon ve kısıtlara göre uygunluğu araştırılır. Buna göre en uygun popülasyonun çoğalmasına izin verilir. Bu bireyler çaprazlama işlemi sonunda çocuk adı verilen yeni bireyler üretirler. Çocuk kendisini meydana getiren ebeveynlerin özelliklerini taşır. Böylece iyi özelliğe sahip olan bireylerin yayılması sağlanır. Probleme ait en iyi çözümün bulunabilmesi için;

- Bireylerin gösterimi doğru bir şekilde yapılmalı,
- Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
- Doğru genetik işlemciler seçilmelidir.

Genetik algoritmalar, diğer eniyileme yöntemleri kullanılırken büyük zorluklarla karşılaşılan, oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedir. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler. Ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler

bulurlar. Genetik algoritmaların asıl amacı, hiçbir çözüm tekniği bulunmayan problemlere çözüm aramaktır. Genetik algoritmalar ancak;

- Arama uzayının büyük ve karmaşık olduğu,
- Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu,
- Problemin belirli bir matematiksel modelle ifade edilemediği,
- Geleneksel eniyileme yöntemlerinden istenen sonucun alınmadığı alanlarda

etkili ve kullanışlıdır.

Buna göre genetik algoritma, başlangıç populasyonu oluşturulması, çaprazlama, mutasyon ve seçim aşamalarını içerir.

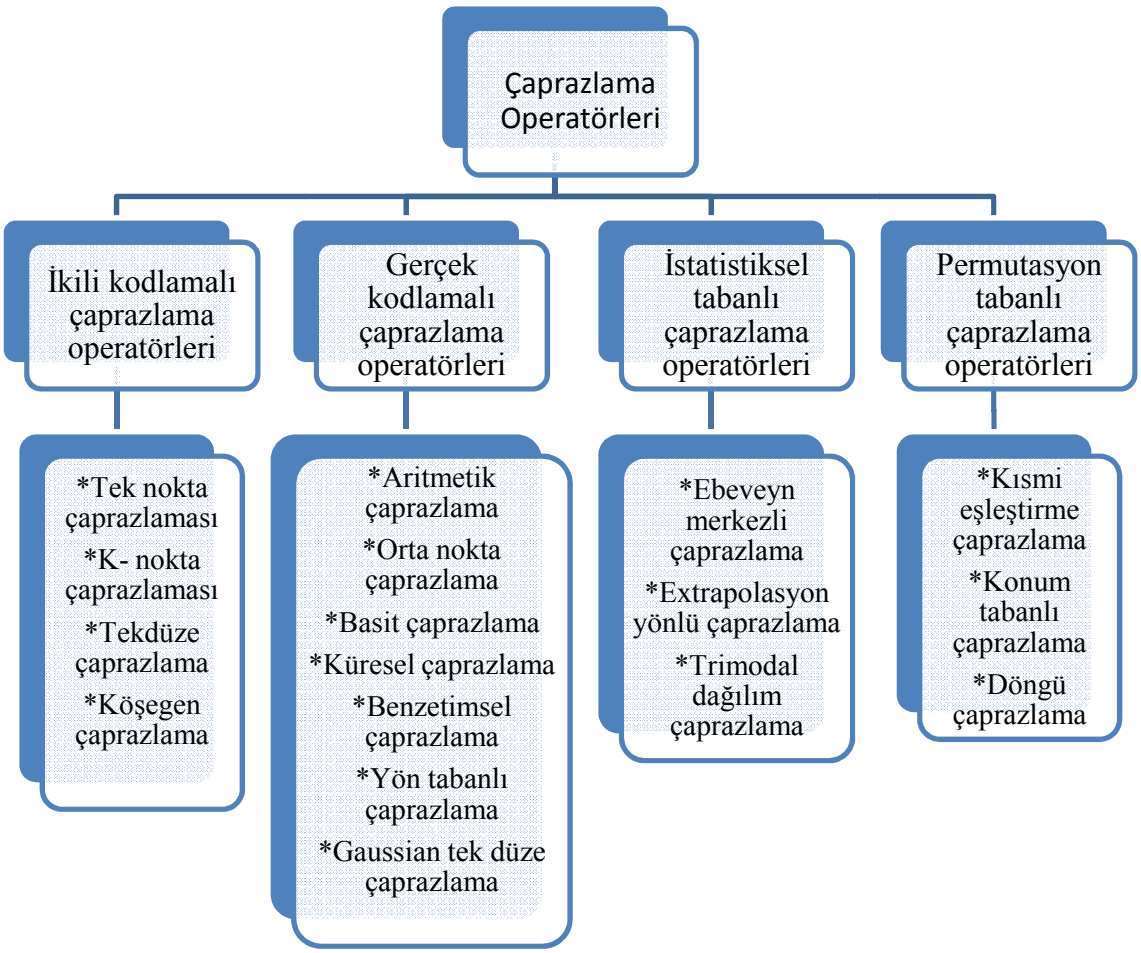
Bu operatörlerden mutasyon operatörü, bir kromozomdaki bir yada daha fazla gen değerinin değişikliğe uğramasını sağlamaktadır. Bu sayede populasyondaki kromozom çeşitliliği devamlı olarak korunmakta ve populasyonun tek bir yerel en iyiye yakınsama sorunu ortadan kalkmaktadır. Kromozomların yapısına göre beş ana mutasyon operatörü mevcuttur. Bunlardan, bit değiştirme operatörü 0-1 gen türlerinde kullanılan ve genlerin değerlerini zıt sayıya çeviren bir mutasyon operatörüdür. Tamsayılı veya float tipi genlere uygulanan mutasyon operatörleri ise sınır, standart olmayan, standart ve Gaussian mutasyon operatörleridir. Bunlardan biri olan sınır operatörü, seçilen genin değerini, genin alt veya üst sınırı olan değere dönüştürür. Bu sınırların seçimi tamamiyle rassal olarak gerçekleştirilmektedir. Tekdüze olmayan mutasyon operatörü, jenerasyon sayısı artarken mutasyon miktarını sıfıra yaklaştıran olasılığı artırmaktadır. Bu sayede evrimin erken safhalarında populasyonu durgunlaştırmaktan kurtarır ve ileri safhalarda genetik algoritmanın iyileşmesini sağlamaktadır. Tekdüze mutasyon operatörü ise seçilen genin değerini, o gen için kullanıcı tarafından seçilen alt ve üst sınırlar arasında rassal olarak seçilen bir değere çevirmektedir. Gaussian mutasyon operaötürnde seçilen genin değerine Gaussian dağılan bir değeri eklemektedir. Gen değerinin kullanıcı tarafından seçilen sınırlar dışında kalması durumunda değer sınırlar arasında getirilmektedir.

Genetik algoritma operatörlerinden bir diğeri olan çaprazlama operatörleri, bilginin kromozomlar arasında aktarılmasını sağlamaktadırlar. Bu işlemi iki ebeveynin bilgilerini harmanlayıp yeni bir çocuk kromozom oluşturarak gerçekleştirmektedir.

Belli bir süre sonra bu çocuklardan bir tanesinin ebeveyn kromozomların en iyi özellikleri aldığı görülür ki bu da bizi çözüme ulaştıran önemli faktörlerden biridir. Çaprazlama operatörleri 4 ana başlıkta incelenebilmektedir.

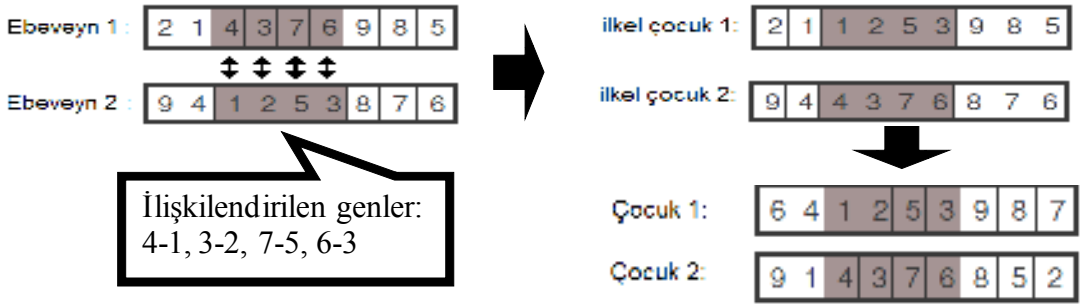
- İkili kodlamalı çaprazlama operatörleri
- Gerçek kodlamalı çaprazlama operatörleri
- İstatistiksel tabanlı çaprazlama operatörleri
- Permutasyon problemleri için çaprazlama operatörleri

Bu 4 ana başlık altında yer alan çaprazlama çeşitleri Şekil 4.1'de başlıklar halinde verilmiştir. İzleyen bölümde problem ile ilgili olması nedeniyle permutasyon problemlerine yönelik çaprazlama ayrıntılı olarak açıklanacaktır.



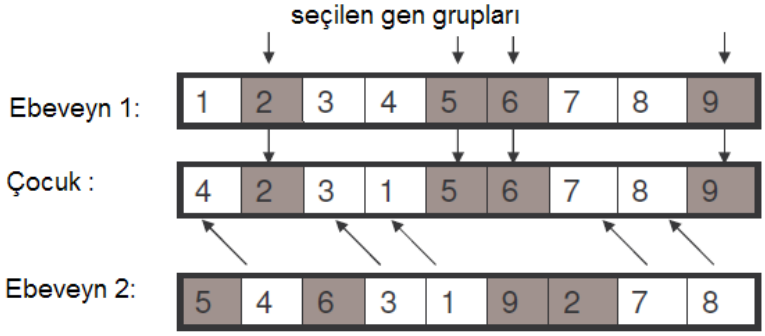
Şekil 4.1. Çaprazlama operatörleri

Permutasyon tabanlı gösterimler için çaprazlama operatörlerinden ilki olan kısmi eşleştirme çaprazlama, tüm pozisyonların çocuk kromozomda sadece bir adet bulunmasını garanti etmektedir. Bu çaprazlama için öncelikle iki adet kromozom ve iki adet çaprazlama noktası rassal olarak seçilir. Bu çaprazlama noktaları arasında kalan genler karşılıklı olarak eşleştirilir ve eğer çaprazlama noktaları dışında kalan genlerde bu genlere rastalınırsa eşgenle değiştirilir. Ancak bu durumda aynı kromozomda bazı genlerin tekrarlanması ya da bazı genlerin silinmesi söz konusu olmaktadır. Bu nedenle kromozomlar üzerinde düzeltmeler yapılarak doğru kromozom yapısına tekrar ulaşılır (Kellegöz vd. 2008).



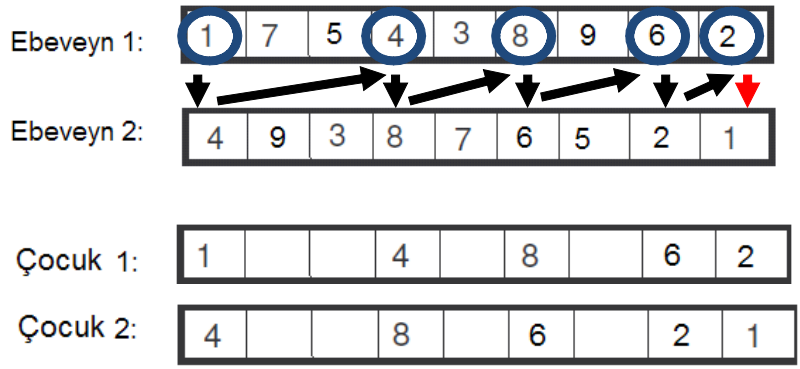
Şekil 4.2. Kısmi eşleştirme çaprazlaması

Permutasyon tabanlı gösterimelere yönelik olan bir diğer çaprazlama türü de konum tabanlı çaprazlamadır. Bu çaprazlamada, bir ebeveynden rassal olarak bir gen grubu seçilir. Bu gen grupları aynı konumda olacak şekilde çocuk kromozoma aktarılmakta ve aktarılan bu genler eş kromozomdan silinmektedir. Daha sonra çocuk kromzomdaki boş genler, eş kromozomdaki gen sırasına göre doldurulur (Kellegöz vd. 2008).



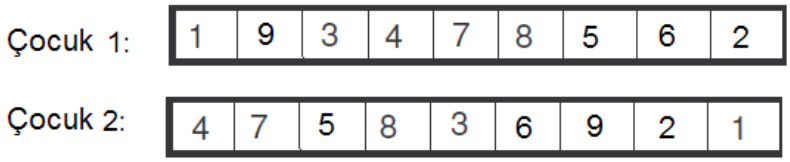
Şekil 4. 3. Konum tabanlı çaprazlama

Döngü (cycle) çaprazlamada ise ilk ebeveyndeki ilk sıradaki gen seçilir. İkinci gendeki aynı sıradaki gen, ilk ebeveynde sabitlenir. Bu genle aynı sırada olan 2. ebeveyndeki gen, ilk ebeveynde sabitlenir. Bu işlem ilk ebeveynde sabitlendiği halde tekrar sabitlenmeye çalışıldığı duruma kadar tekrarlanır. Şekil 4. 4'de verilen örnekte 1 nolu gen sabitlendiği halde tekrar sabitlenmeye çalışıldığı için işlem sonlandırılır. Aynı işlem 2. ebeveyn için de gerçekleştirilir. Buna göre çocuk kromozomların ilk durumları Şekil 4. 4'de gösterilmiştir.



Şekil 4. 4. Döngü çaprazlama

Daha sonra boş kalan alanlar diğer kromozomdaki aynı konumdaki genlerle doldurulur . (Şekil 4. 5) (Kellegöz vd. 2008)



Şekil 4. 5. Döngü çaprazlama

Çaprazlama işlemi ile bir işin diğer popülasyonda hangi tezgahta ve hangi sırada olacağı bilgisi ilgili popülasyona kopyalanır.

Buna göre problemimizde iş kromozomlarındaki genler sıralı ve yenilenmeyen özellik gösterdikleri için yer tabanlı çaprazlama yöntemi uygulanacaktır. Diğer çaprazlama yöntemlerinin uygulanması sonucunda bazı genlerin tekrarlanması veya bazı genlerin ise kromozomdan silinmesi mümkün olabilecektir.

4.2.2. Geliştirilen genetik algoritma

Buna göre problemin çözümünde kullanılacak genetik algoritmada kullanılacak prosedür ve operatörler aşağıdaki gibidir.

1. Başlangıç popülasyonu oluştur.
2. Hesapla
3. Seçim
4. Çaprazlama (sıralı çaprazlama)
5. Mutasyon (iki çeşit)

Buna göre programın adımları aşağıdaki gibidir.

```
BAŞLA
        İlk nesli oluştur

        For  g=1 to jenerasyon_sayısı
                Uyum değerlerini hesapla
                seçim
                For i=1 to popülasyon_sayisi
                        Çaprazlama
                        Mutasyon
                Next
                Genişletilmiş popülasyon oluştur
        Next
SON
```

4.2.2.1. Kromozom yapısı ve ilk neslin türetilmesi

Kromozomlar, iş sıralamasını ve tezgahların başlangıç ve bitiş noktalarını gösteren iki alt kromozomdan oluşmaktadır. İş sıralamasını gösteren kromozomda iş sayısı (21) kadar gen bulunur ve bu genlere birbirinden farklı olacak şekilde rassal olarak değer atanır. Tezgah kromozomu ise tezgah adedi(4 adet) ilgili tezgaha atanacak iş sayısını göstermektedir. Buna göre, Şekil 4. 6'de bir tek populasyona ait olan tezgah ve iş kromozomlarının yapısı gösterilmiştir. Şekil 4. 7'de 21 adet işin olduğu iş kromozomu yer almaktadır. Buna göre işler genlere tezgahlara atanacak sırada yerleştirimektedir. Şekil 4. 8'de 4 tezgahlık bir örnek için tezgah kromozomu gösterimine yer verilişitir. Buna göre, 1 nolu tezgah 7 adet iş, 2 ve 3 nolu tezgahlarda

4'er adet iş ve 4 nolu tezgahta ise 6 adet işin atanacağı gösterilmektedir. Şekil 4. 9'de ise bu iki kromozomun birbirleri üzerindeki etkileri gösterilmiştir. 4 adet tezgah ve 21 adet işin olduğu biraz önce değinilen örnekte verilen kromozomlara göre iş kromzomundaki ilk 7 adet iş sırasıyla 1 nolu tezgaha atanmakta, sonraki 4 adet iş 2 nolu tezgah, ardından gelen 4 adet iş 3 nolu tezgaha ve arda kalan 6 adet iş ise sırasıyla 4 nolu tezgaha atanmaktadır.

tezgah (p,1)	tezgah (p,2)	…	tezgah (p,4)	İskromozomu (p,1)	İskromozomu (p,2)	…	İskromozomu (p,21)

Şekil 4. 6. Bir populasyona ait kromozom gösterimi

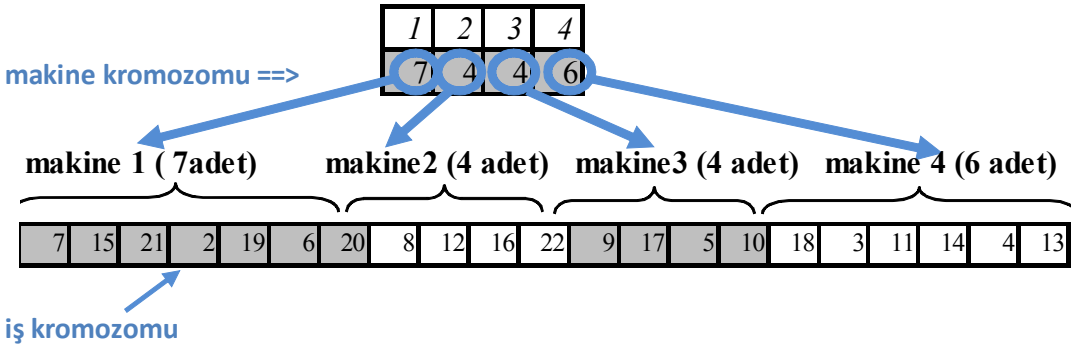
Örnek:

7	15	21	2	19	6	20	8	12	16	22	9	17	5	10	18	3	11	14	4	13

Şekil 4. 7. Bir populasyona ait iş kromozom gösterimi

1	*2*	*3*	*4*
7	4	4	6

Şekil 4. 8. Bir populasyona ait tezgah kromozomu gösterimi



Şekil 4. 9. Tezgah ve iş kromozomlarının birlikte vei taşıdıkların anlamların gösterimi

İlk nesil türetilirken, atamanın yapılacağı tezgahta işlerin yapılıp yapılmamasına göre atama gerçekleştirilmiştir. Tezgahlara iş ataması ancak ilgili işin o tezgahta yapılabiliyor olması durumunda gerçekleştirilmektedir. Böylece kısıtlara uygun

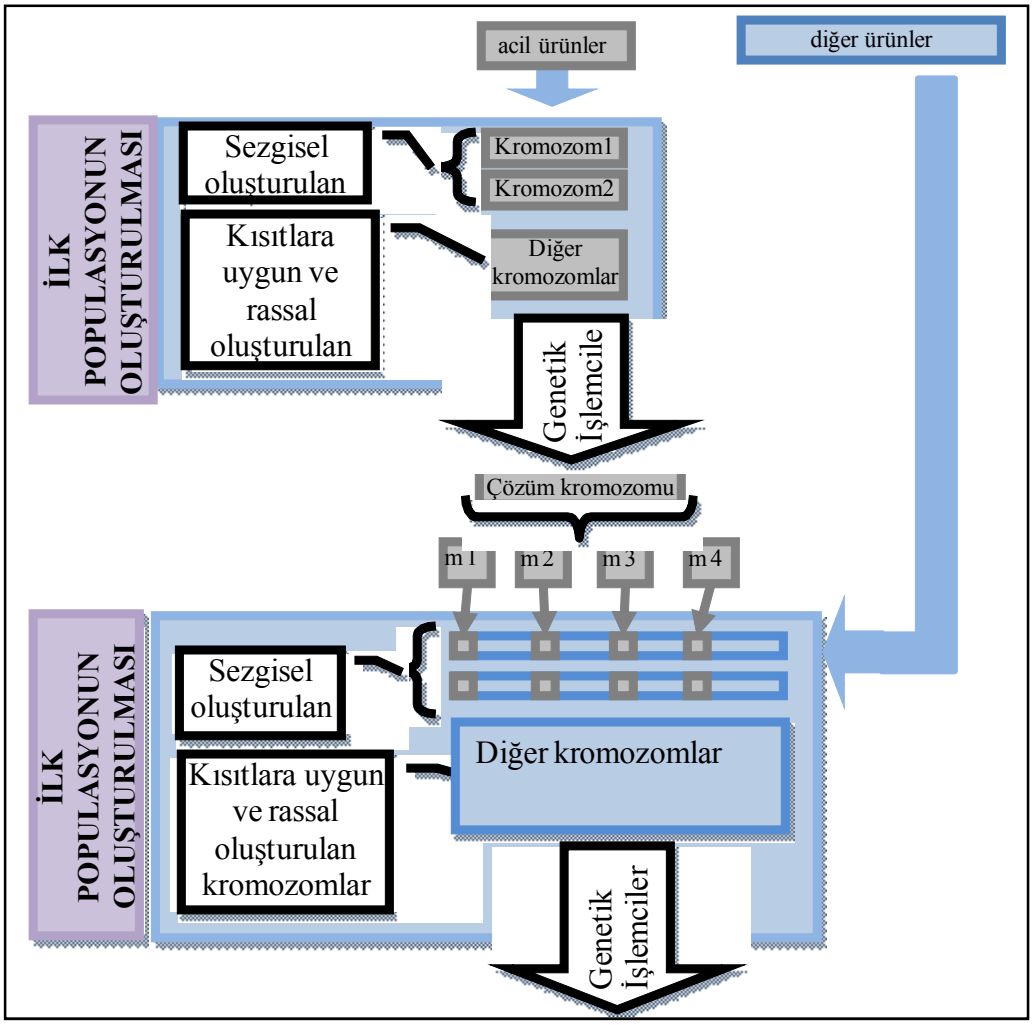
olmayan atamalar baştan hiç yapılmayarak çözüm sırasında uygun olmayan kromozomlar üzerinde işlem yapılması sonucu oluşan zaman kaybının önüne geçilmiş olacaktır.

İlk neslin türetilmesinde uygulanan diğer bir yöntem ise ilk 2 kromozom atamalarının sezgisel tabanlı olarak oluşturulmasıdır. Buna göre ilk iki kromozomun ataması sırasında iki adım kullanılmıştır.

İlk adımda "0" teslim zamanlı işlerin en iyilemesi yapılmıştır. Bunun için yine ilk iki kromozomun sıralaması mantıksal tabanlı gerçekleştirilmiştir. Buna göre önceki işe göre en az hazırlık ve işlem süresi gerektiren iş, sıradaki gene atanmıştır. Daha sonra bu kromozomlar içinde iki farklı tür iyileştirme yapılmıştır. İlk tür iyileştirmede en fazla iş yüküne sahip tezgahın son işi en az iş yüküne sahip tezgahtaki en az hazırlık ve işlem süresi gerektiren yere ataması gerçekleştirilmiştir. Diğer iyileştirmede ise en fazla iş yüküne sahip tezgahın son işi en az iş yüküne sahip tezgahtaki en az hazırlık ve işlem süresi gerektiren yere atanması durumunda bu iki tezgahın en fazla iş yükü miktarı azalıyorsa, bu atama gerçekleştirilir. Eğer bir azalma sağlanamazsa farklı bir tezgah ve eğer o ürün için hiç bir değişiklik gerçekleştirilemezse de bir önceki işe geçilir. Böylece "0" teslim zamanlı işler için ilk iki kromozomun ataması gerçekleştirilmiş olmaktadır. Sonraki kromozomların ataması, işin ilgili tezgahta yapılabilmesi koşuluyla tamamen rassal olarak gerçekleştirilir. Oluşturulan bu kromozomlar genetik algoritma işlemlerine tabi tutularak çözüm elde edilir.

İkinci adımda ise önceki adımda belirlenen sıralama sabitlenerek sonraki atamalar yine en az hazırlık ve işlem süresi gerektirecek işin atanması suretiyle gerçekleştirilir. Bu atamalar gerçekleştirildikten sonra "0" teslim zamanlı işlerin sıralamasını değiştirmeden önceki adımda anlatılan iyileştirme işlemleri gerçekleştirilir. Böylece ilk iki kromozomun ataması bu iki adımla tamamlanmış olur. Daha sonra diğer kromozomların ataması, işin ilgili tezgahta yapılabilmesi koşuluyla tamamen rassal olarak gerçekleştirilir. Oluşturulan bu kromozomlar genetik algoritma işlemlerine tabi tutularak çözüm elde edilir. Bu sayede nispeten daha iyi bir başlangıç noktasından çözüme başlanmış olmaktadır. Tezgah kromozomları için ise toplam iş sayısının tezgahlara olabildiğince eşit dağıtılması suretiyle, her tezgaha yüklenecek iş yükünün

olabildiğince yakın olması sağlanmıştır. İlk neslin oluşturulmasında izlenen adımlar Şekil 4. 10'da gösterilmiştir.

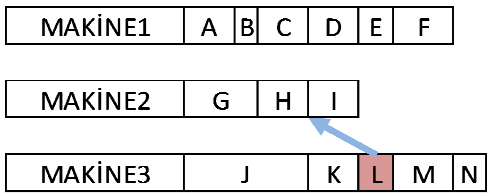


Şekil 4. 10. Geliştirilen genetik algoritmada ilk nesil oluşturma adımları

4.2.2.2. Mutasyon

Önerilen GA'da 2 tür mutasyon uygulanmaktadır. İlk mutasyonda; iş yükü en fazla olan tezgahtaki en fazla gecikmeye sahip ve diğer tezgahta (en az iş yüküne sahip tezgah) yapılabilecek olan iş seçilip, aktarılacak olan tezgahta gecikmesi olmayacak bir noktaya atanır. Buna bağlı olarak işi azaltılan en fazla iş yüküne sahip tezgahın, tezgah kromozomundaki geninde yer alan iş adeti 1 eksiltilirken, diğer tezgah genindeki iş sayısı 1 arttırılır. Örneğin, Şekil 4. 11'deki "L" işi enyüksek iş yüküne sahip tezgahtaki

en fazla gecikmesi olan iş olsun. Bu iş en az iş yükü olan 2 nolu tezgahta gecikmesi olmayan en son nokta olan 3. sıraya atanacaktır.

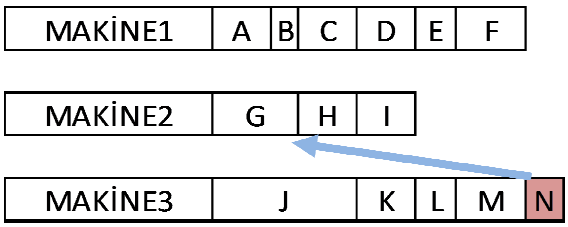


Şekil 4. 11. I. Tip mutasyon gösterimi

I. yöntemin adımları:

1. İş yükü en fazla olan tezgahtaki en fazla gecikmeye sahip olan ve en az iş yüküne sahip tezgahta yapılabilecek iş seçilir
2. Aktarılacak tezgahta gecikmesi olmayacak bir noktaya atanır.
3. Tezgah kromozomundaki işin aktarıldığı tezgah genindeki sayı 1 artırılır.
4. Tezgah kromozomundaki işin alındığı tezgah genindeki sayı ise 1 azaltılır.

İkinci tip mutasyonda ise en yüksek iş yüküne sahip tezgahtaki son iş seçilir. Eğer bu iş, en düşük iş yüküne sahip tezgahta yapılabiliyorsa bu tezgahtaki en az hazırlık ve işlem süresi gerektiren sıraya atanır. Ancak yer değişimi yapılacak iş aktarılacak tezgahta yapılamıyorsa veya bir sonraki en düşük iş yüküne sahip tezgahta arama işlemi yapılır. Buradaki en az hazırlık ve işlem süresine ihtiyaç duyacak olan sıra tezgahlardaki en yüksek iş yükünü azaltıyorsa bu sıraya atama yapılır. İlgili iş diğer hiçbir tezgahta yapılamıyorsa en yüksek iş yüküne sahip tezgahta bulunan bir önceki işe geçilir. Örneğin, Şekil 4. 12'de, en fazla iş yüküne sahip olan tezgah 3 nolu tezgahtır ve bu tezgahtaki en son iş ise "N" işidir. Bu iş en düşük iş yüküne sahip 2 nolu tezgahta yapılabiliyorsa bu tezgahtaki en az hazırlık süresi gerektiren noktaya atanır. Ancak 2 nolu tezgahta yapılamıyorsa bir sonraki en düşük iş yüküne sahip tezgah olan 1 nolu tezgahtaki en uygun sıra belirlenir. Bu iş diğer hiçbir tezgahta yapılamıyorsa N işinden önce yer alan "M" işine geçilip adımlar tekrarlanır.



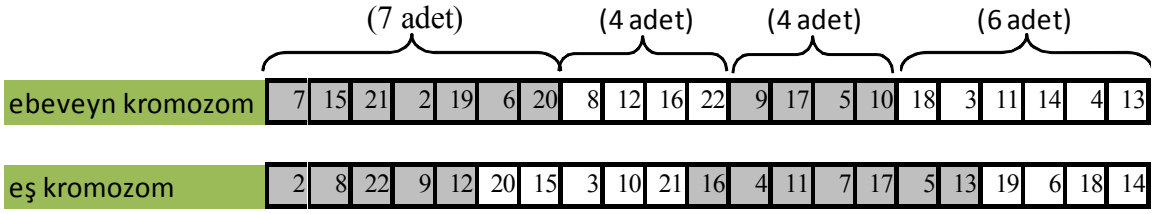
Şekil 4. 12. II.Tip mutasyon gösterimi

II.Yöntem adımları

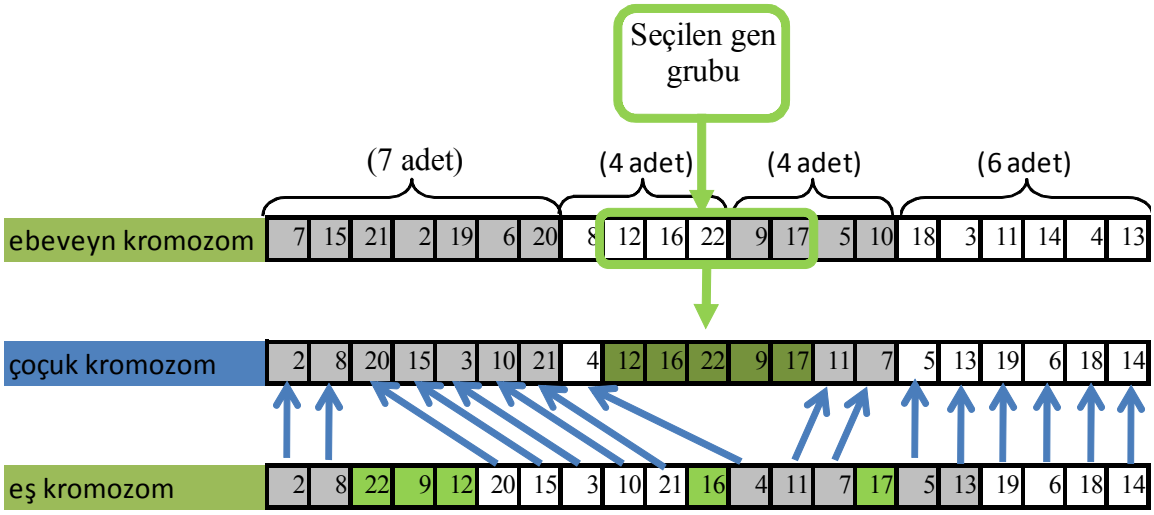
1. En fazla iş yüküne sahip tezgahtaki son iş seçilir
2. Eğer bu iş en az iş yüküne sahip tezgahta yapılabiliyorsa bu tezgahtaki en az hazırlık süresi gerektiren sıra belirlenir. Ancak yapılamıyorsa bir sonraki en az iş yüküne sahip tezgahtaki en az hazırlık süresi gerektiren yer belirlenir.
3. Eğer bu sıraya atama yapılmasıyla, çizelgelemedeki en yüksek iş yükü azalıyorsa bu atama gerçekleştirilir.
4. İlgili iş diğer hiçbir tezgahta yapılamıyorsa en fazla iş yüküne sahip tezgahtaki bir önceki işe geçilerek adımlar tekrarlanır.

4.2.2.3. Çaprazlama

Geliştirilen genetik algoritmada konum tabanlı çaprazlama yöntemi ugulanmıştır. Bu çaprazlamada bir kromozomdan rassal olarak bir gen grubu seçilir. Bu gen grupları, aynı pozisyonda olacak şekilde çocuk kromozoma aktarılır. Çocuk kromozomdaki boş genlere eş kromozomdaki gen sıralamasına göre atama yapılır. Örneğin çaprazlama uygulanacak kromozomdan rassal olarak 9-13 nolu genler arası seçilmiş olsun. Buna göre Şekil 4. 13'de çaprazlama öncesi ebeveyn kromozomlar, Şekil 4. 14'de çaprazlama işlemine yönelik şekil ve Şekil 4. 15'de çaprazlama sonrası elde edilen çocuk kromozomlar gösterilmiş olup çaprazlama adımları Şekil 4. 16'de verilmiştir.



Şekil 4. 13. Çaprazlama öncesi ebeveyn kromozomlar



Şekil 4. 14. Çaprazlama işlemi

Şekil 4. 15. Çaprazlama sonrası çocuk kromozom

BAŞLA

0 ile 1 arasında rassal bir sayı seç

If rassal sayı < çaprazlama olasılığı **then**

{Çaprazlama noktası olarak Rassal olarak 2 adet nokta seçilir}

{Eş kromozom seçilir}

{Çaprazlama uygulanacak kromozom için bu iki nokta arasındaki genler doğrudan çocuk kromozoma aktarılır}

{çocuk kromozomdaki bu iki noktanın dışındaki genler ise diğer kromozomdaki sıraya göre atanır}

End if

DUR

Şekil 4. 16. Çaprazlama işlemi adımları

4.2.2.4. Seçim yöntemi

Mutasyon ve çaprazlamaya uğrayarak oluşturulmuş çocuk kromozomlar ve ebeveyn kromozomlar arasında amaç fonksiyonu değerine göre sıralama yapılır. Bu sıralamadan, popülasyon büyüklüğüne ulaşılıncaya kadar kromozomlar sırasıyla yeni popülasyona aktarılır.

4.3. Matematiksel Model İçin Kullanılan Çözücüler

Geliştirilen matematiksel model'in yedi adet gerçek-hayat problemi için LINGO 6.0 ve GAMS eniyileme araçlarıyla elde edilen sonuçları araştırılmıştır. Bu eniyileme araçlarından LINGO 6.0, büyük ölçekli problemlerin kısaca modellenmesini, çözülmesini ve sonuçların analiz edilmesini ve sonuç olarak doğrusal ve doğrusal olmayan eniyilemenin gücünü kullanmamızı sağlayan basit bir araçtır. LINGO 6.0, farklı tipteki modellerin çözümü için doğrudan çözücü, doğrusal çözücü, doğrusal olmayan çözücü ve dal-sınır yöneticisi olmak üzere 4 çözücü içerir.

Model çalıştırıldığında, doğrudan çözücü ilk olarak olabildiğince çok değişkenin değerlerini hesaplar. Eğer sadece bir tek bilinmeyen değişkenli eşitlik kısıtı bulursa, o değişkenin kısıt sağlayan değerini belirler. Doğrudan çözücü, bilinmeyen değişkenler tükenince veya bilinmeyen değişkene sahip daha başka herhangi bir eşitlik kısıtı kalmayınca sonlanır. Doğrudan çözücü durduğunda tüm değişkenler hesaplanmışsa, LINGO çözüm raporunu oluşturur. Eğer bilinmeyen değişkenler mevcutsa, LINGO modelin yapısını ve matematiksel içeriğini inceleyerek hangi çözücünün modelde kullanılacağına karar verir. Yani, sürekli doğrusal modeller için doğrusal çözücüyü, bir veya daha fazla doğrusal olmayan kısıtın olduğu modeller için ise doğrusal olmayan çözücüyü çalıştırır. Modelin herhangi bir tamsayılı değişken içermesi durumunda ise dal-sınır yöneticisi devreye girer ve modelin doğasına göre ya doğrusal ya da doğrusal olmayan çözücüleri çağırır. LINGO'daki doğrusal çözücü revize edilmiş simplex metodunu (revised simplex method with product form inverse) kullanır. LINGO'nun doğrusal olmayan çözücüsü ise hem arıdşık doğrusal programlama (successive linear programming (SLP)) hem de genelleştirilmiş azaltılmış gradient algoritmasını (generalized reduced gradient (GRG) algorithms) çalıştırır. Bariyer çözücüsü ise doğrusal modellerin çözümünde seçeneğe bağlı olarak sağlanabilir. Tamsayılı modeller dal-sınır metodu kullanılarak çözülürken, doğrusal tamsayılı modellerde ise, LINGO önemli ön işlemler gerçekleştirir. Örneğin, tamsayılı olmayan uygun çözüm alanını sınırlandırmak için kesici kısıtlar ekler. Bu kesici kısıtlar çoğu tamsayılı modellerde çözüm süresini büyük ölçüde iyileştirir.

Çözümde kullanılan diğer bir eniyileme aracı olan GAMS (genel cebirsel modelleme sistemi- General Algebraic Modelling System) içinde birçok çözücü

seçeneği bulunduran bir arayüz görevini gören bir yazılım paketidir. GAMS Development Corporation tarafından kurulmuş olup, bu firmanın internet sitesinden birçok yararlı bilgiye ulaşılabilir (GAMS, 2010)

GAMS modelleme ve eniyileme problemlerinin (doğrusal, doğrusal olmayan ve karma tamsayılı) çözümü için kullanılan yüksek seviyeli bir programlama dilidir. Özellikle büyük ölçekli karmaşık problemlerin modellenmesinde çok kullanışlıdır. İlk olarak optimizasyon uzmanları Dr. Anthony Brooke and Dr. Alexander Meeraus tarafından dünya bankası için geliştirilmiştir. The computer science technical section of the operations research society of america tarafından 1987 yılında ödül alan programın en temel özelliği güçlü olmasının yanı sıra kullanım kolaylığı ve esnekliğidir. CPLEX, DICOPT, OLS, CONOPT, MINOS, XA ve MILES gibi çözücüleri içinde bulunduran GAMS, bu çözücüler sayesinde doğrusal, doğrusal olmayan ve karma tamsayılı modelleri çözebilmekte ve kullanıcının çözüm yöntemini kolaylıkla değiştirebilmesine olanak sağlamaktadır.

GAMS programında kullanılan CPLEX çözücüsü, doğrusal modelleri birçok farklı algoritma kullanarak çözmektedir. Doğrusal modellerin büyük bir çoğunluğu CPLEX'in değiştirilmiş (modified) ilkil (primal) simpleks algoritması kullanılarak çözülmektedir. Belirli tipteki problemler, aternatif ikil simplex algoritması, ağ eniyileme yada bariyer algoritmasının kullanılmasından fayda elde etmektedir. Doğrusal modellerin çözümü hafıza yoğun olmaktadır. CPLEX hafızayı çok verimli bir şekilde yönetmesine rağmen yetersiz fiziksel hafıza büyük ölçekli modellerin çözümünde en yaygın sorunlardan biridir. Hafızanın yetersiz olduğu durumlarda CPLEX performansı olumsuz yönde etkileyebilen bazı ayar düzeltmelerini otomatikman yapmaktadır. CPLEX , varsayılan ayarlamaları kullanarak doğrusal modellerin çoğunu çözmek için tasarlanmıştır. Bu ayarlamalar genellikle eniyi hız ve güvenilirliği sağlamakla beraber, bazı problemlerin performanslarını arttırmak için bu ayarlar değiştirilebilmektedir. Bazı problemler, varsayılan metod olan ilkil simplex algoritmasına göre ikil simplex algoritması ile daha hızlı çözülebilmektedir. Özelikle, sağ taraf katsayılarındaki küçük değişkenliğe ve maliyet katsayılarında yüksek derecede değişkenliğe sahip olan büyük ölçüde dejenere olmuş modeller ikil simpleks

kullanılarak çok daha hızlı çözülebilmetedirler. Hatta hem ilkil hem de ikil simpleks metodlarının ikisiyle de çok zayıf performans gösteren az sayıda problem de mevcuttur.

GAMS/Cplex ek olarak uygun olmayan çözümü tespit etme (Cplex infeasibility bulucu) imkanı tanımaktadır. Bu infesibility bulucu uygun olmayan doğrusal modeli almakta ve basitleştirilemeyen kararsız kısıt kümeleri üretmektedir (IIS: irreducibly inconsistent set of constraints). IIS uygun olmayan, ancak kümeden bir elemanın çıkarılmasıyla uygun çözüme dönüşen bir kısıt ve değişken sınırları kümesidir .

Tamsayılı yada karma tamsayılı modellerin çözümünde kullanılan metodlar doğrusal modellerin çözümünde kullanılan metodara göre çok daha fazla matematiksel hesaplama gerektirmektedir. Bir tek karma tamsayılı model birçok doğrusal alt problem türetmesinden dolayı Cplex bu modeller için kesme kısıtlı dal-sınır algoritmasını kullanmaktadır (Utexas, 2010)

4.4. Problem Seti

Önerilen çözüm yaklaşımlarının performanslarını görebilmek için farklı boyutlarda 8 adet gerçek-hayat problemi GAMS-Cplex, LINGO 6.0 ve geliştirilen açgözlü genetik algoritma ile çözülmüştür. Ele alınan bu 8 gerçek hayat problemi tezgah sayıları 3 ile 8 adet , iş sayıları ise 7 ile 382 adet aralığında değişen bu gerçek hayat problemlerindeki işlem süreleri 0,375 ile 4,65 dakika arasında değişirken, hazırlık süreleri ise 0 ile 3,7 dakika arasında değişmektedir. Tablo 4.2'deki problem boyutu sütunundan da görülebileceği gibi ele alınan problemlerin olası çözüm sayısının alt sınırı 40320 ile $447{,}335*10^{835}$ arasında değişmekte olup her problemde farklı teslim zamanları olan problem setleri ele alınmıştır. Çizelgeleme işlemi sırasında havuzda bekleyen işlerle henüz yapılmamış işler tekrar sıralama işlemine tabi tutulmaktadır. Bu nedenle, işlerin tezgahlara atanması işleminin yapıldığı sırada, herhangi bir tezgahtaki işin bitmesi durumunda ilgili tezgahın boş bekleyeceği gerçeğiyle çizelgeleme işleminin en fazla 10 dakika içinde tamamlanması istenmektedir. Aksi takdirde tezgahların boş bekleme süresince kaybedilen zaman üretim verimliliğini olumsuz etkileyecektir. Bu nedenle, tüm çözücülere, çözüm için 10 dakikalık bir süre verilmiş ve 10 dakikalık süre sonunda ulaşılan en iyi değerler Tablo 4.2'ye yansıtılmıştır.

Tablo 4.1. Geliştirilen açgözlü genetik algoritmada kullanılan parametreler

	G.A. Parametreleri
populasyon sayısı	50
mutasyon olasılığı	1
çaprazlama olasılığı	0,5

Tablo 4.2. Ele alınan problemler ve özellikleri

Test örnekleri	Tezgah sayısı	İş sayısı	Hazırlık süresi aralığı	İşlem süresi aralığı	Problem boyutu	0 teslim zamanlı iş sayısı	x teslim zamanlı iş sayısı
A	3	7	0-3,7	1,15-3,4	40320	7	0
B	3	25	0,9-3,7	1,15-3,4	$4,03291*10^{26}$	25	0
C	3	30	0,9-3,7	1,85-4,275	$8,22283*10^{33}$	20	10 (x=10)
D	4	182	2,1-3,7	0,8-4,275	$218,96*10^{336}$	10	172 (x=120)
E	5	220	0,9-3,7	0,375-3,4	$2468,98*10^{425}$	10	210 (x=100)
F	6	240	0,9-3,7	0,425-4,275	$140,02*10^{476}$	20	220 (x=100)
G	8	320	0,9-3,7	0,375-4,275	$241,0696*10^{677}$	15	305 (x=110)
H	8	382	0-3,7	0,475-4,65	$447,335*10^{835}$	20	362 (x=120)

4.5. Çözüm Sonuçları

Farklı boyutlarda seçilen test problemlerinin özellikleri Tablo 4.2'de ve GAMS-CPLEX, LINGO ve geliştirilen genetik algoritma ile çözülmesiyle elde edilen çözümler Tablo 4. 3'de gösterilmiştir. Buna göre, 10 dakikalık zaman kısıtının bulunması nedeniyle, sadece 40320 olası çözümü olan en küçük boyutlu problemde (A problemi)

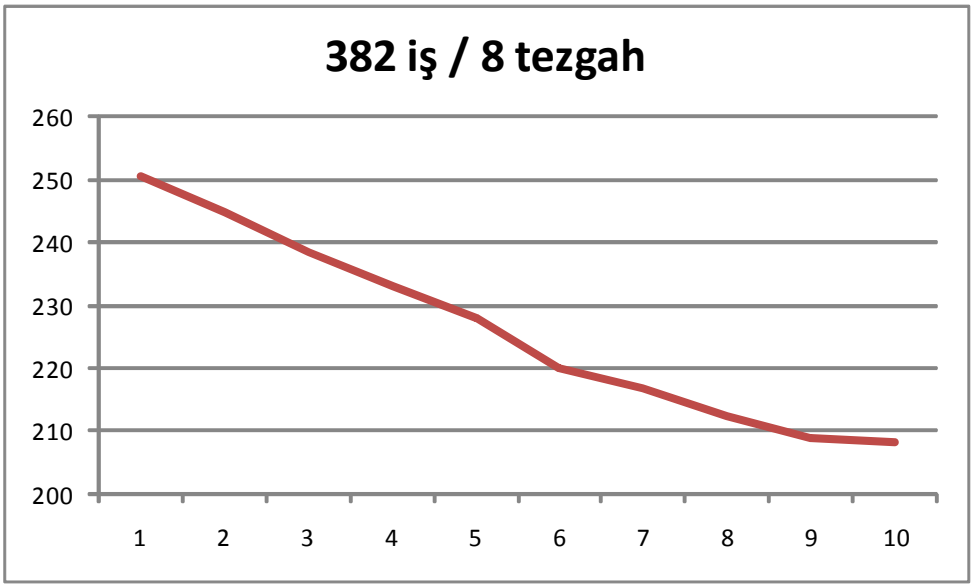
genetik algoritma ve GAMS-Cplex çözücüsü aynı sonuca ulaşmıştır. Bu problemde işlem süresi aralığı 1,15 ile 3,4 dakika arasında değişen işler ve hazırlık sürelerinin aralığı da 0 ile 3,7 dakika arasında değişen hazırlık işlemlerinin yer aldığı 7 işin ve 3 tezgahın bulunduğu ve tüm işlerin teslim zamanlarının "0" olduğu bir problem ele alınmıştır. Tablo 4.3-B'de çözüm sonuçları verilen ikinci problemimiz 25 adetlik iş ve 3 adet tezgahın oluşturduğu ve $4*10^{26}$ olası çözümünün bulunduğu bir problem ele alınmıştır. Hazırlık süreleri 0-9 ve 3,7 dakika arasında ve işlem sürelerinin 1,15 ile 3,4 arasında değiştiği bu problemin çözümünde GAMS-Cplex ve LINGO 6.0 çözücüleri izin verilen süre olan 10 dakikada uygun bir çözüme ulaşmalarına rağmen geliştirilen genetik algoritmanın bulduğu çözüme ulaşamamışlardır. Benzer sonuç, 30 işin ve 3 tezgahın yer aldığı 3. örnek problemde de karşımıza çıkmakta ve bu örnekte de GAMS-Cplex ve LINGO 6.0 çözücüleri geliştirilen genetik algoritmanın performans bakımından gerisinde kalmaktadırlar. Bu problemlerin dışında kalan örneklerde ise izin verilen süre olan 10 dakikalık süre içnde GAMS-Cplex ve LINGO 6.0 çözücüleri herhangi bir uygun çözüme dahi ulaşamazken geliştirilen genetik algoritma Tablo 4.3 de verilen sonuçlara ulaşmıştır. Geliştirilen açgözlü genetik algoritmanın çözüme ulaşma sürecindeki performansını sunmak amacıyla Şekil 4. 17'de 382 adet işin ve 8 adet tezgahın bulunduğu "G" örnek probleminin 10 dakikalık süre içinde ulaştığı değerlerin grafiksel gösterimine yer verilmiştir. Mutasyon ve çaprazlama işlemleri sonucu oluşan çocuk kromozomlar ile ebeveyn kromozomlar içinden en iyilerinin bir sonraki jenerasyona aktarılmasından dolayı tüm kromozomlara mutasyon uygulanmasıyla açgözlü bir genetik algoritma uygulanmış olup mutasyon olaslığı bu nedenle 1 olarak belirlenmiştir. Geliştirilen bu genetik algoritmanda kullanılan parametreler ise Tablo 4.1'de verilmiş olup 50 adetlik populasyon büyüklüğü, 0,5 çaprazlama olasılığ ve 1 mutasyon olasılığı seçilmiştir.

Tablo 4. 3. Farklı çözüm yöntemleri ile ulaşılan sonuçlar

		A	B	C	D	E	F	G	H
Çözücüler	Geliştirilen G.A.	31,75	81,675	129,78	476,75	175,38	278,1	25,45	208,38
	Gams-Cplex	31,75	94,039 (*)	185 (*)	(**)	(**)	(**)	(**)	(**)
	LINGO 6.0	48,25 (*)	390,3 (*)	411,68 (*)	(**)	(**)	(**)	(**)	(**)

(*) En iyi çözümü bulamamasına karşın ulaşılan mevcut en iyi uygun çözüm alınmıştır.
(**) 10 dakikalık süre içinde herhangi bir uygun çözüme ulaşamamıştır.

Sonuç olarak, problem boyutunun artmasıyla eniyileme araçları, geliştirilen genetik algoritmanın ulaştığı sonuca göre oldukça yetersiz kalmaktadır. Hatta iş yoğunluğunun fazla olduğu "D", "E", "F", "G", "H" problemlerinde GAMS ve LINGO eniyileme araçları herhangi bir uygun çözüme dahi ulaşamamıştır.



Şekil 4. 17. Problem H için geliştirilen genetik algoritmanın 10 dakikalık zaman diliminde aldığı değerler

SONUÇ VE ÖNERİLER

Bu çalışmada, paralel çok tipli tezgahları yükleme ve üretim partilerini sıralama problemi ele alınmıştır. Daha sonra probleme yönelik matematiksel model geliştirilmiş olup, bu modelin GAMS-CPLEX, LINGO6.0 ile çözüm performansları araştırılmıştır. Sistemin ihtiyacından dolayı çizelgeleme işleminin 10 dakika gibi kısa bir sürede bitmesi gerekliliği ve özellikle eniyileme modellerinin büyük ölçekli modellerin çözümünde etkisiz kalması nedeniyle bu problem genetik algoritma yaklaşımı ile incelenmiştir.

Genetik algoritma ile problem çözülürken, rassal olarak oluşturulan başlangıç popülasyonu, elde edilecek çözümü etkilemesi nedeniyle oldukça önem taşımaktadır. Bu nedenle, ilk nesil iki farklı aşamada oluşturulmuştur. Öncelikle acil işler geliştirilen genetik algoritma adımlarına tabi tutulmuş, elde edilen acil işlere ait çizelgeleme, problem geneline ait olan ilk populasyonun oluşturulması aşamasında sabitlenip, kısıtlara uygun olarak diğer işlerin genlere atanmasıyla ilk aşama tamamlanmış olmuştur. Bu ilk aşamada uygulanan bir diğer yöntem ise ilk iki kromozomdaki gen atamalarının en yakın komşu sezgiseline göre gerçekleştirilmesi, diğer kromozomların ise kısıtlara uygun olacak şekilde rassal olarak atanmasıdır. Bu sayede uygun olmayan kromozomlar üzerinde işlem yapılmayarak zaman kaybı da önlenmiş olmaktadır. Sonuçta, eniyileme araçları sadece en küçük boyutlu örnek problemde geliştirilen genetik algoritmanın bulduğu sonuca ulaşmış olup, daha büyük problemlerde, geliştirilen genetik algoritmanın ulaştığı sonuçtan oldukça uzak değerler elde etmiştir.

Çalışmanın devamında genetik algoritmanın belli bir süre sonunda ulaştığı amaç fonksiyonu değerinin, matematiksel modellerin amaç fonksiyonu değerine üst sınır olarak atanmasıyla eniyileme yöntemlerinin performansının arttırılması sağlanabilir. Ek olarak, büyük ölçekli tamsayılı problemlerin çözümünde kullanılan ayrıştırma yöntemleri de bu tarz problemlerde araştırılabilecek bir başka konu olarak karşımıza çıkmaktadır.

KAYNAKLAR DİZİNİ

Alok Singh, Anurag Baghel, 2009, A new grouping genetic algorithm approach to the multiple traveling salesperson problem, Soft Computing - A Fusion of Foundations, Methodologies & Applications ,13-1, p95-101, 7p.

Angel RD, Caudle WL, Noonan R, Whinston A., 1972, Computer assisted school bus scheduling. Management Science, 18, p279–288.

Applegate David L., Bixby Robert E., Chvátal Vasek , Cook William J., 2007, The Traveling Salesman Problem, Princeton University Press.

Basu A, Elnagar A, Al-Hajj A., 2000, Efficient coordinated motion.Mathematical and Computer Modelling, 31, p39–53.

Dantzig GB, Fulkerson DR, Johnson SM., 1954, Solution of a large-scale traveling salesman problem. Operations Research , 2, p393–410.

Behnamian J., Zandieh M., Ghomi S.M.T. Fatemi ., 2009, Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. Expert Systems with Applications: An International Journal, 36-6 , p9637-9644

Brown EC, Ragsdale CT, Carter AE., 2007, A grouping genetic algorithm for the multiple traveling salesperson problem. International Journal of Information Technology and Decision Making., 06-2 , p333-347

Brummit B, Stentz A., 1996, Dynamic mission planning for multiple mobile robots. Proceedings of the IEEE international conference on robotics and automation.

Calvo RW, Cordone R. 2003, A heuristic approach to the overnight security service problem. Computers and Operations Research, 30, p1269–1287.

Chen A. L., Yang G. K. and Wu Z. M., 2008, Production scheduling optimization algorithm for the hot rolling processes; International Journal of Production Research, 46- 7, p1955–1973.

Christofides N, Mingozzi A, Toth P., 1981, Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Mathematical Programming, 20, p255–282.

Clark Christopher M., Morton R., and Bekey G.A., 2008, "Altruistic Relationships for Optimizing Task Fulfillment in Robot Communities" Distributed Autonomous Robot Systems 8, p261-270.

Fröhlich R. and Hüsig M., 2009, Job Batching and Scheduling for Parallel Non-Identical Machines via MILP and Petri Nets, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA.

GAMS (2010), http://www.gams.com, Erişim 10 Mart 2010.

Gavish B.,1976, A note on the formulation of the *m*salesman traveling salesman problem. Management Science, 22(6), p704–705.

Gilbert KC, Hofstra RB., 1992, A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem. Decision Sciences ,23, p250–259.

Huang, Simin; Cai, Linning; Zhang, Xiaoyue, 2010, Parallel dedicated machine scheduling problem with sequence-dependent setups and a single server, Computers & Industrial Engineering (0360-8352), 58-1, p165-174.

Jing Qian, Pessan, C., Nguyen Huynh Tuong; Neron, E., 2009, Computers & Industrial Engineering, 2009. CIE 2009. International Conference on (978-1-4244-4135-8). 6-9. p7.

Kara, İ., Bektaş, T., 2006, "Integer Programming Formulations of Multiple Salesmen Problems and its Variations". European Journal of Operational Research , 174-3, p1449-1458.

Kellegöz T., Toklu B. and Wilson J., 2008, Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem, Applied Mathematics and Computation, 199-2, p590-598.

Kim KH, Park Y., 2004, A crane scheduling method for port container terminals. European Journal of Operational Research ,156, p752–768.

Laporte G, Nobert Y., 1980, A cutting planes algorithm for the *m*-salesmen problem. Journal of the Operational Research Society , 31, p1017–1023.

Laporte G., 1992, The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research, 9, p345–358.

Miller CE, Tucker AW, Zemlin RA., 1960, Integer programming formulation of traveling salesman problems. Journal of Association for Computing Machinery,7, p326–329.

Lenstra JK, Rinnooy Kan AHG., 1975, Some simple applications of the traveling salesman problem. Operational Research Quarterly, 26, p717–733.

L. Li, F. Qiao, and Q.D. Wu, 2009, ACO-Based Scheduling of Parallel Batch Processing Machines to Minimize the Total Weighted Tardiness, 5th Annual IEEE Conference on Automation Science and Engineering Bangalore, India, p280-285.

Macharis C, Bontekoning YM., 2004, Opportunities for OR in intermodal freight transport research: a review. European Journal of Operational Research , 153, p400–416.

Mateus RochadePaula, Geraldo Robson Mateus, Martín Gómez Ravetti , 2010, A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times .Computers & Operations Research (0305-0548) ,37-5, p938-949.

Naderi, B. Ruiz, Ruben, Zandieh, M., 2010, Algorithms for a realistic variant of flowshop scheduling, Computers & Operations Research (0305-0548), 37-2, p.236-246.

Okonjo-Adigwe C., 1988, An effective method of balancing the workload amongst salesmen. Omega, 16-2, p159–163.

Ryan J.L., Bailey T.G., Moore J.T and Carlton W.B., 1998, Reactive tabu search in unmanned aerial reconnaissance simulations, Proceedings of the 1998 winter simulation conference, 1, p873–879.

Saleh HA and Chelouah R., 2004, The design of the global navigation satellite system surveying networks using genetic algorithms, Engineering Applications of Artificial Intelligence, 17, p111–122.

Svestka JA and Huckfeldt VE., 1973, Computational experience with an m-salesman traveling salesman algorithm. Management Science ,19-7, p790–799.

Tang L, Liu J, Rong A and Yang Z., 2000, A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex. European Journal of Operational Research, 124, p267–282.

Utexas (2010), Http://www.che.utexas.edu/cache/gams.html, Erişim 10 Mart 2010.

Wang X. and Regan AC., 2002, Local truckload pickup and delivery with hard time window constraints. Transportation Research Part B, 36, p97–112.

Weimin Liu , Sujian Li , Aiyun Zheng , Fanggeng Zhao, 2009, Tobacco distribution vehicle routing program and the resolving method, Proceedings of the 21st annual international conference on Chinese control and decision conference, p.5208-5211.

Xiang W. and Lee, H.P., 2008, Ant colony intelligence in multi-agent dynamic manufacturing scheduling, Engineering Applications of Artificial Intelligence 21, p73–85.

Ying, Kuo-Ching, Cheng, Hui-Miao, Ying, Kuo-Ching., 2010, Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic, Expert Systems with Applications (0957-4174), 37-4, p.2848-2852.

Yu Z, Jinhai L, Guochang G, Rubo Z. and Haiyan Y, 2002, An implementation of evolutionary computation for path planning of cooperative mobile robots, Proceedings of the fourth world congress on intelligent control and automation, 3, p1798–1802.

Zhi-Long Chen , Warren B. Powell, 1999, Solving Parallel Machine Scheduling Problems byColumn Generation,.INFORMS Journal on Computing, 11-1, p78 - 94.

Zhicong Zhang, Li Zheng, Michael X. Weng, 2007, Dynamic parallel machine scheduling with mean weighted tardiness objective by Q-Learning .The International Journal of Advanced Manufacturing Technology, 34 , p968-980.

Zhenzhen Xu, Feng Xia, and Xianchao Zhang., 2009, Multi-Robot Dynamic Task Allocation Using Modified Ant Colony System, Artificial Intelligence and Computational Intelligence, p288-297.

Zne-Jung Lee, Shih-Wei Lin, Kuo-Ching Ying., 2010, Scheduling jobs on dynamic parallel machines with sequence-dependent setup times, Int J Adv Manuf Technol, 47, p773–781.

EKLER

EK1. Komax Tezgahı Üretim Standart Zaman Tablosu

Time table Cutting Standard wire till 3,0 mm² base on Komax 433

Time table

length mm	time min/100
200	1,3
700	1,4
1200	1,5
1700	1,6
2200	1,7
2700	1,8
3200	1,9
3800	2,0
4300	2,1
4800	2,2
5300	2,3
5800	2,4
6300	2,5
6800	2,6
7300	2,7
7900	2,8
8400	2,9
8900	3,0
9400	3,1
9900	3,2
10400	3,3
10900	3,4
11400	3,5
11900	3,6
12000	3,7
K 433-1.mdd	

length mm	time min/100
400	1,6
900	1,7
1400	1,8
2000	1,9
2500	2,0
3000	2,1
3500	2,2
4000	2,3
4500	2,4
5000	2,5
5500	2,6
6000	2,7
6500	2,8
7000	2,9
7500	3,0
8000	3,1
8500	3,2
9000	3,3
9500	3,4
10000	3,5
10500	3,6
11000	3,7
11500	3,8
12000	3,9
K 433-2.mdd	

length mm	time min/100
400	1,8
500	1,9
600	1,9
900	1,9
1500	2,0
2000	2,1
2600	2,2
3100	2,3
3700	2,4
4200	2,5
4800	2,6
5300	2,7
5900	2,8
6400	2,9
7000	3,0
7500	3,1
8100	3,2
8600	3,3
9200	3,4
9700	3,5
10300	3,6
10800	3,7
11400	3,8
11900	3,9
12000	4,0
K 433-3.mdd	

length mm	time min/100
200	2,0
700	2,1
1200	2,2
1700	2,3
2300	2,4
2800	2,5
3300	2,6
3800	2,7
4400	2,8
4900	2,9
5400	3,0
5900	3,1
6500	3,2
7000	3,3
7500	3,4
8000	3,5
8600	3,6
9100	3,7
9600	3,8
10100	3,9
10700	4,0
11200	4,1
11700	4,2
12000	4,3
K 433-4.mdd	

length mm	time min/100
100	2,0
600	2,1
1100	2,2
1700	2,3
2200	2,4
2700	2,5
3200	2,6
3800	2,7
4300	2,8
4800	2,9
5400	3,0
5900	3,1
6400	3,2
7000	3,3
7500	3,4
8000	3,5
8500	3,6
9100	3,7
9600	3,8
10100	3,9
10700	4,0
11200	4,1
11700	4,2
12000	4,3
K 433-5.mdd	

length mm	time min/100
500	2,2
1000	2,3
1500	2,4
2000	2,5
2500	2,6
2600	2,7
3000	2,7
3500	2,8
4000	2,9
4500	3,0
4900	3,1
5400	3,2
5900	3,3
6400	3,4
6900	3,5
7400	3,6
7900	3,7
8400	3,8
8900	3,9
9400	4,0
9900	4,1
10400	4,2
10900	4,3
11400	4,4
11900	4,5
12000	4,6
K 433-6.mdd	

Base settings / parameter for all variants

Speed	100-80%
Acceleration	90%

Press optimation Station 1	on
Press optimation Station 2	on

Wait time Station 1	100 ms
Wait time Station 2	100 ms

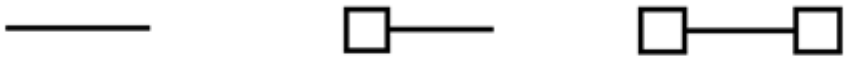
Time covers the run time for cutting wires. No setup include.

Time is in min/100 without allowance

EK2. Yacc Tezgahı Üretim Standart Zaman Tablosu

Wire Cutting Standard base on Yacc 7

Time table



length mm	time min/100	length mm	time min/100	length mm	time min/100
200	1,5	4300	4,3	8300	7,0
300	1,6	4500	4,4	8400	7,1
500	1,7	4600	4,5	8600	7,2
600	1,8	4700	4,6	8700	7,3
800	1,9	4900	4,7	8900	7,4
900	2,0	5000	4,8	9000	7,5
1100	2,1	5200	4,9	9200	7,6
1200	2,2	5300	5,0	9300	7,7
1400	2,3	5500	5,1	9500	7,8
1500	2,4	5600	5,2	9600	7,9
1700	2,5	5800	5,3	9700	8,0
1800	2,6	5900	5,4	9900	8,1
1900	2,7	6000	5,5	10000	8,2
2100	2,8	6100	5,5	10200	8,3
2200	2,9	6200	5,6	10300	8,4
2400	3,0	6400	5,7	10500	8,5
2500	3,1	6500	5,8	10600	8,6
2700	3,2	6700	5,9	10800	8,7
2800	3,3	6800	6,0	10900	8,8
3000	3,4	7000	6,1	11100	8,9
3100	3,5	7100	6,2	11200	9,0
3300	3,6	7200	6,3	11400	9,1
3400	3,7	7400	6,4	11500	9,2
3600	3,8	7500	6,5	11700	9,3
3700	3,9	7700	6,6	11800	9,4
3900	4,0	7800	6,7	12000	9,5
4000	4,1	8000	6,8	YEL 10 Yacc 7_12000mm.mdd	
4200	4,2	8100	6,9		

Base setting/parameter for all variants

Speed					

Time covers the run time for cutting wires. No setup include.

Time is in min/100 without allowance

EK3.GAMS MODELİ

```
sets
i/i1*i382/
m/m1*m8/
;
alias(i,j);

parameter KALANSUREG(i)/
$call =xls2gms "i=C:\Documents and Settings\ASRA\My Documents\prog\23.XLSm"
o=C:\KALANSUREG.inc R=KALANSUREG
$include C:\KALANSUREg.inc
/;
parameter SIPARISG(i)/
$call =xls2gms "i=C:\Documents and Settings\ASRA\My Documents\prog\23.XLSm"
o=C:\SIPARISG.inc R=SIPARISG
$include C:\SIPARISG.inc
/;
table ISLEMSURESIG(i,m)
$call =xls2gms "i=C:\Documents and Settings\ASRA\My Documents\prog\23.XLSm"
o=C:\ISLEMSURESIG.inc R=ISLEMSURESIG
$include C:\ISLEMSURESIG.inc
;
table HAZIRLIKG(i,j)
$call =xls2gms "i=C:\Documents and Settings\ASRA\My Documents\prog\23.XLSm"
o=C:\HAZIRLIKG.inc R=HAZIRLIKG
$include C:\HAZIRLIKG.inc
;
binary variables
sira(i,j,m)
;
positive variables
bitis(i,m)
skotu(i,m)
siyi(i,m)
u(i,m)
;

equations
smobj
sm1(i)
sm2(j)
sm3(i,m)
sm4(m)
sm5(m)
sm6(i,m)
sm7(i,j,m)
sm8(i,m)
sm9(i,j,m)
sm10(i,m)
;
free variable zsm;

smobj..zsm=e=sum((i,m),skotu(i,m));
sm1(i)..sum((j,m)$(ord(j)<>ord(i)),sira(i,j,m))=e=1;
sm2(j)..sum((i,m)$(ord(i)<>ord(j)),sira(i,j,m))=e=1;
```

```
sm3(i,m)..sum(j$(ord(i)<>ord(j)),sira(i,j,m))-sum(j$(ord(i)<>ord(j)),sira(j,i,m))=e=0;
sm7(j,i,m)$(ord(i)<>ord(j) and ord(j)>card(m))..
bitis(j,m)=g=bitis(i,m)+(HAZIRLIKG(i,j)+ISLEMSURESIG(j,m)*SIPARISG(j))*sira(i,j,m)+10000000*
(sira(i,j,m)-1);
sm6(i,m)$(ord(i)=ord(m))..sum(j$(ord(i)<>ord(j)),sira(i,j,m))=e=1;
sm8(i,m)$(ord(i)>card(m))..bitis(i,m)+siyi(i,m)-
skotu(i,m)=e=KALANSUREG(i)*sum(j$(ord(j)<>ord(i)),sira(i,j,m));
sm9(i,j,m)$((ord(i)>card(m))and(ord(j)>card(m))AND (ord(i)<>ord(j)))..u(i,m)-u(j,m)+(card(i)-
card(m)+1)*sira(i,j,m)=l=card(i)-card(m);
option mip=cplex;
sonmodel.optcr=0;

model sonmodel/sm1,sm2,sm3,sm6,sm7,sm8,sm9,smobj/;

OPTION RESLIM = 600;

solve sonmodel minimizing zsm using mip;
display sira.l;
display bitis.l;
display skotu.l;
```

EK4. LINGO modeli

```
MODEL:
SETS:
PARCA/1..382/;
MAKINE/1..8/;
KUME1(PARCA,PARCA,MAKINE):SIRA;
KUME2(PARCA):SIPARISL,KALANSUREL;
KUME3(PARCA,PARCA):HAZIRLIKL;
KUME4(PARCA,MAKINE):BITIS,KOTU,IYI,ISLEMSURESIL,U;
ENDSETS

DATA:
KALANSUREL=@OLE('C:\Documents and Settings\ASRA\My Documents\prog\382.XLSm');
SIPARISL=@OLE('C:\Documents and Settings\ASRA\My Documents\prog\382.XLSm');
ISLEMSURESIL=@OLE('C:\Documents and Settings\ASRA\My Documents\prog\382.XLSm');
HAZIRLIKL=@OLE('C:\Documents and Settings\ASRA\My Documents\prog\38.XLSm');
ENDDATA

MIN=@SUM(PARCA(I)|I#GE#4:@SUM(MAKINE(M):KOTU(I,M)));

@FOR(MAKINE(M):
  @FOR(PARCA(I)|I#EQ#M:
        @SUM(PARCA(J)|J#NE#I:SIRA(I,J,M))=1));

@FOR(PARCA(I):
        @SUM(PARCA(J)|J#NE#I:@SUM(MAKINE(M):SIRA(I,J,M)))=1);

@FOR(PARCA(J):
        @SUM(PARCA(I)|I#NE#J:@SUM(MAKINE(M):SIRA(I,J,M)))=1);

@FOR(PARCA(I):
        @FOR(MAKINE(M):
                @SUM(PARCA(J)|J#NE#I:SIRA(I,J,M))-
@SUM(PARCA(J)|J#NE#I:SIRA(J,I,M))=0));

@FOR(PARCA(J)|J#GE#4:
 @FOR(MAKINE(M):
  @FOR(PARCA(I)|I#NE#J:
        BITIS(J,M)>=BITIS(I,M)+(HAZIRLIKL(I,J)+ISLEMSURESIL(J,M)*SIPARISL(J))*SIRA(I,J,
M)+10000000*(SIRA(I,J,M)-1))));


@FOR(PARCA(I):
        @FOR(MAKINE(M):
                BITIS(I,M)+IYI(I,M)-
KOTU(I,M)=KALANSUREL(I)*@SUM(PARCA(J)|J#NE#I:SIRA(I,J,M))));


@FOR(MAKINE(M):
  @FOR(PARCA(I)|I#GE#4:
        @FOR(PARCA(J)|J#NE#I #and# J#GE#4:U(I,M)-U(J,M)+375*SIRA(I,J,M)<=374)));


@FOR(KUME1:@BIN(SIRA));

END
```

EK5. Genetik Algoritma Kodları

```
Dim gecicikr() As Variant
Dim gecicim() As Variant
Dim toplamisyuk(), hazir(), ortceza, sureilk(), adim As Variant
Dim ceza() As Variant
Dim misadet() As Integer
Dim ilkgrup As Integer
Dim kromozom() As String
Dim u, pop_sayisi, m_sayisi, p As Integer
Dim enyuksek(), enyuksekno(), endusuk(), endusukno() As Variant
Dim bitis() As Variant
Dim dizino() As Variant
Dim islemsuresi() As Variant
Dim kalansure() As Variant
Dim sip() As Variant
Dim ilkatama() As Variant
Dim hazirlik() As Variant

Public Sub GA()
Dim d, gen_size As Integer
Dim thetamax As Variant
Dim m As Integer
ilkgrup = Range("ilkgrup")
m_sayisi = Range("mak_sayisi").Value
u = Range("is_adedi").Value
pop_sayisi = Range("pop_sayisi").Value

ReDim sureilk(1 To m_sayisi) As Variant
ReDim gecicikr(1 To 2, 1 To u) As Variant
ReDim gecicim(1 To 2, 1 To m_sayisi) As Variant
ReDim ilkatama(1 To m_sayisi) As Variant


For adim = 1 To 2
   If adim = 1 Then
      gen_size = 20
   Else
      gen_size = Range("gensize").Value
   End If

      initialpop
      hesapla
      eniyibelirle

      For g = 1 To gen_size
      Sheets("anasayfa").Cells(11, 2) = g
         For p = 2 To pop_sayisi
            crossover
            r = Rnd
            If r < 0.5 Then
            iyilestirmax
            Else
               mutation

            End If
         Next
            hesapla
```

```
            eniyibelirle
        Next
        sabitle

Next
End Sub

Public Sub initialpop()
Dim z, i, j, jj, s, y, atanmis As Integer

ReDim kromozom(1 To pop_sayisi, 1 To u) As String
ReDim kume(1 To u) As Integer
ReDim a(1 To u) As Integer
ReDim misadet(1 To pop_sayisi, 1 To m_sayisi) As Integer
For p = 1 To 3
mantiksalmaxyuk
Next
basla = 4

For p = basla To pop_sayisi
If adim = 1 Then

   For s = 1 To ilkgrup - m_sayisi
      kume(s) = s + m_sayisi
   Next
   sayi = ilkgrup - m_sayisi
Else
   For s = 1 To u - m_sayisi
      kume(s) = s + m_sayisi
   Next
   sayi = u - m_sayisi

End If
If adim = 1 Then
      For m = 1 To m_sayisi
         misadet(p, m) = Int((ilkgrup - m_sayisi) / m_sayisi)
      Next
      kalan = (ilkgrup - m_sayisi) Mod m_sayisi
      For b = 1 To kalan
         misadet(p, m_sayisi - b) = misadet(p, m_sayisi - b) + 1
      Next
Else
      For m = 1 To m_sayisi
         misadet(p, m) = Int((u - m_sayisi) / m_sayisi)
      Next
      kalan = (u - m_sayisi) Mod m_sayisi
      For b = 1 To kalan
         misadet(p, m_sayisi - b) = misadet(p, m_sayisi - b) + 1
      Next

End If
   j = 0
   For m = 1 To m_sayisi

      For k = 1 To misadet(p, m)
         j = j + 1
            z = Int(sayi * Rnd) + 1
```

```
            Do While Range("ISLEMSURESI").Cells(kume(z), m) = 100
               z = Int(sayi * Rnd) + 1
            Loop
            kromozom(p, j) = kume(z)

            For x = z + 1 To sayi
               kume(x - 1) = kume(x)
            Next
            sayi = sayi - 1
      Next
   Next
Next
End Sub


Public Sub mutation()
Dim bas, bit, farkno As Integer
Dim fark As Variant

mutprob = Range("mutprob").Value
r = Rnd
If r < mutprob Then
   bit = 0

   For k = 1 To enyuksekno(p)
      bit = bit + misadet(p, k)
   Next
   bas = bit - misadet(p, enyuksekno(p))
   fark = bitis(p, kromozom(p, bas + 1)) - Range("kalansure").Cells(kromozom(p, bas + 1),
enyuksekno(p))
   farkno = bas + 1

   If endusukno(p) <= Range("yaccsayisi") Then
         For i = 1 To misadet(p, enyuksekno(p))
               N = bitis(p, kromozom(p, bas + i)) - Range("kalansure").Cells(kromozom(p, bas + i),
enyuksekno(p))
               If fark < N And Range("ISLEMSURESI").Cells(kromozom(p, bas + i), endusukno(p)) <>
100 Then
                  fark = bitis(p, kromozom(p, bas + i)) - Range("kalansure").Cells(kromozom(p, bas + i),
enyuksekno(p))
                  farkno = bas + i
               End If

         Next
      bit = 0
      bas = 0
         For k = 1 To endusukno(p)
            bit = bit + misadet(p, k)
         Next
         bas = bit - misadet(p, endusukno(p))
         isl = Range("ISLEMSURESI").Cells(kromozom(p, farkno), endusukno(p)) *
Range("SIPARIS").Cells(kromozom(p, farkno))
         fark = Range("KALANSURE").Cells(kromozom(p, farkno)) - (isl +
Range("HAZIRLIK").Cells(1, Int(kromozom(p, farkno))))
         If fark < 0 Then
            ata = bas + 1
         Else
```

```
            For i = bas + 2 To bit
               fark = Range("KALANSURE").Cells(kromozom(p, farkno)) - (bitis(p, kromozom(p, i - 1)) +
Range("HAZIRLIK").Cells(Int(kromozom(p, i - 1)), Int(kromozom(p, farkno))) + isl)
               If fark < 0 Then
                  ata = i - 1
                  Exit For
               End If

            Next
          End If
          If ata = 0 Then
            ata = bit
          End If
                        aktarurun = kromozom(p, farkno)
                        f = Abs(farkno - ata)

                        If farkno > ata Then
                           For j = 1 To f - 1
                              kromozom(p, farkno - j + 1) = kromozom(p, farkno - j)
                           Next
                           kromozom(p, ata + 1) = aktarurun
                        Else
                           For j = 1 To f
                              kromozom(p, farkno + j - 1) = kromozom(p, farkno + j)
                           Next
                           kromozom(p, ata) = Int(aktarurun)
                        End If
   Else
          For i = 1 To misadet(p, enyuksekno(p))
               N = bitis(p, kromozom(p, bas + i)) - Range("kalansure").Cells(kromozom(p, bas + i),
enyuksekno(p))
               If fark < N Then
                  fark = bitis(p, kromozom(p, bas + i)) - Range("kalansure").Cells(kromozom(p, bas + i),
enyuksekno(p))
                  farkno = bas + i
               End If

          Next

          bit = 0
          bas = 0

          For k = 1 To endusukno(p)
            bit = bit + misadet(p, k)
          Next
          bas = bit - misadet(p, endusukno(p))

          isl = Range("ISLEMSURESI").Cells(kromozom(p, farkno), endusukno(p)) *
Range("SIPARIS").Cells(kromozom(p, farkno))
          fark = Range("KALANSURE").Cells(kromozom(p, farkno)) - (isl +
Range("HAZIRLIK").Cells(1, Int(kromozom(p, farkno))))
          If fark < 0 Then
            ata = bas + 1
          Else

            For i = bas + 2 To bit
```

```
            fark = Range("KALANSURE").Cells(kromozom(p, farkno)) - (bitis(p, i - 1) +
Range("HAZIRLIK").Cells(Int(kromozom(p, i - 1)), Int(kromozom(p, farkno))) + isl)
            If fark < 0 Then
               ata = i
               Exit For
            End If

          Next
        End If
        If ata = 0 Then
          ata = bit
        End If

                    aktarurun = kromozom(p, farkno)
                f = Abs(farkno - ata)

                If farkno > ata Then
                   For j = 1 To f - 1
                      kromozom(p, farkno - j + 1) = kromozom(p, farkno - j)
                   Next
                   kromozom(p, ata + 1) = aktarurun
                Else
                   For j = 1 To f
                      kromozom(p, farkno + j - 1) = kromozom(p, farkno + j)
                   Next
                   kromozom(p, ata) = Int(aktarurun)
                End If


   End If
      misadet(p, endusukno(p)) = misadet(p, endusukno(p)) + 1
      misadet(p, enyuksekno(p)) = misadet(p, enyuksekno(p)) - 1
tekhesapla
End If
End Sub

Public Sub hesapla()
ReDim ceza(1 To pop_sayisi) As Variant
Dim toplamisyuk(), hazir() As Variant
ReDim toplamisyuk(1 To pop_sayisi, 1 To m_sayisi) As Variant
Dim p, oncekiis, iss, j As Variant
ReDim bitis(1 To pop_sayisi, 1 To u) As Variant
ReDim enyuksek(1 To pop_sayisi), hazir(1 To pop_sayisi), enyuksekno(1 To pop_sayisi), endusuk(1 To
pop_sayisi), endusukno(1 To pop_sayisi) As Variant


Dim a, z As Variant

For p = 1 To pop_sayisi
   hazir(p) = 0

   enyuksek(p) = -10000
   enyuksekno(p) = 0
   endusuk(p) = 1000000000000#
   endusukno(p) = 0
   j = 1
   For m = 1 To m_sayisi
```

```
   a = 0

      oncekiis = m
      For k = 1 To misadet(p, m)
         iss = Int(kromozom(p, j))
         bitis(p, iss) = a + Range("HAZIRLIK").Cells(oncekiis, iss) + Range("ISLEMSURESI").Cells(iss,
m) * (Range("SIPARIS").Cells(iss))
        a = bitis(p, iss)
         hazir(p) = hazir(p) + Range("HAZIRLIK").Cells(oncekiis, iss)
         oncekiis = iss

         j = j + 1
      Next

      If j > 1 Then
         toplamisyuk(p, m) = bitis(p, kromozom(p, j - 1))
      Else
         toplamisyuk(p, m) = 0
      End If

      If enyuksek(p) < toplamisyuk(p, m) Then
         enyuksek(p) = toplamisyuk(p, m)
         enyuksekno(p) = m

      End If
      If endusuk(p) > toplamisyuk(p, m) Then
         endusuk(p) = toplamisyuk(p, m)
         endusukno(p) = m

      End If
   Next
Next

If adim = 1 Then
jjson = ilkgrup
Else
jjson = u
End If

For p = 1 To pop_sayisi
   ceza(p) = 0

   For i = m_sayisi + 1 To jjson
      If Range("KALANSURE").Cells(i) < bitis(p, i) Then
         ceza(p) = ceza(p) + (bitis(p, i) - Range("KALANSURE").Cells(i))

      End If
   Next
         topceza = topceza + ceza(p)
         adet = adet + 1
Next

ortceza = topceza / adet
End Sub

Public Sub tekhesapla()
Dim toplamisyuk(), hazir() As Variant
```

```
ReDim toplamisyuk(1 To pop_sayisi, 1 To m_sayisi) As Variant
ReDim hazir(1 To pop_sayisi) As Variant
Dim oncekiis, iss, j As Variant


Dim a, z As Variant

   hazir(p) = 0
   enyuksek(p) = 0
   enyuksekno(p) = 0
   endusuk(p) = 100000000000000#
   endusukno(p) = 0
   j = 1
   For m = 1 To m_sayisi
      If adim = 1 Then
   a = 0
   Else
   a = sureilk(m)
   End If


      oncekiis = m
      For k = 1 To misadet(p, m)
               iss = Int(kromozom(p, j))
         bitis(p, iss) = a + Range("HAZIRLIK").Cells(oncekiis, iss) + Range("ISLEMSURESI").Cells(iss,
m) * (Range("SIPARIS").Cells(iss))
         a = bitis(p, iss)
         hazir(p) = hazir(p) + Range("HAZIRLIK").Cells(oncekiis, iss)
         oncekiis = iss

         j = j + 1
      Next
      If j > 1 Then
         toplamisyuk(p, m) = bitis(p, kromozom(p, j - 1))

      Else
         toplamisyuk(p, m) = 0
      End If
      If enyuksek(p) < toplamisyuk(p, m) Then
         enyuksek(p) = toplamisyuk(p, m)
         enyuksekno(p) = m

      End If
      If endusuk(p) > toplamisyuk(p, m) Then
         endusuk(p) = toplamisyuk(p, m)
         endusukno(p) = m

      End If
   Next

If adim = 1 Then
jjson = ilkgrup
Else
jjson = u - ilkgrup + m_sayisi
End If

   ceza(p) = 0
```

```
   For i = m_sayisi + 1 To jjson
      If Range("KALANSURE").Cells(i) < bitis(p, i) Then
         ceza(p) = ceza(p) + (bitis(p, i) - Range("KALANSURE").Cells(i)) + hazir(p)

      End If
   Next
End Sub


Public Sub tekhesaplaher()
Dim son() As Integer
Dim dizi() As Variant
ReDim ceza(1 To pop_sayisi) As Variant
ReDim toplamisyuk(1 To pop_sayisi, 1 To m_sayisi) As Variant
Dim oncekiis, iss, c, j As Variant
ReDim bitis(1 To pop_sayisi, 1 To u) As Variant
ReDim enyuksek(1 To pop_sayisi), hazir(1 To pop_sayisi), enyuksekno(1 To pop_sayisi), endusuk(1 To
pop_sayisi), endusukno(1 To pop_sayisi) As Variant
ReDim son(0 To m_sayisi) As Integer
ReDim dizi(0 To m_sayisi) As Variant
ReDim dizino(0 To m_sayisi) As Variant
Dim a, z As Variant

hazir(p) = 0
enyuksek(p) = 0
enyuksekno(p) = 0
endusuk(p) = 100000000000000#
endusukno(p) = 0
j = 1

For m = 1 To m_sayisi
   If adim = 1 Then
   a = 0
   Else
   a = sureilk(m)
   End If
   If adim = 1 Then
   oncekiis = m
   Else

   oncekiis = ilkatama(m)
   End If
   For k = 1 To misadet(p, m)
      iss = Int(kromozom(p, j))

      bitis(p, kromozom(p, j)) = a + Range("HAZIRLIK").Cells(oncekiis, iss) +
Range("ISLEMSURESI").Cells(kromozom(p, j), m) * (Range("SIPARIS").Cells(kromozom(p, j)))
      c = c + 1
      Worksheets("anasayfa").Cells(17, c + 1) = bitis(p, kromozom(p, j))

      a = bitis(p, kromozom(p, j))
      hazir(p) = hazir(p) + Range("HAZIRLIK").Cells(oncekiis, iss)
      oncekiis = iss
      j = j + 1
   Next
   If j > 1 Then
```

```
        toplamisyuk(p, m) = bitis(p, kromozom(p, j - 1))
        Worksheets("anasayfa").Cells(18, m + 1) = toplamisyuk(p, m)

    Else
        toplamisyuk(p, m) = 0
    End If
Next

For m = 1 To m_sayisi
dizi(m) = toplamisyuk(p, m)
dizino(m) = m
Next
If adim = 1 Then
For m = 1 To m_sayisi
sureilk(m) = toplamisyuk(p, m)
Next
End If
For i = 1 To m_sayisi - 1
    For j = i + 1 To m_sayisi
        If dizi(i) > dizi(j) Then
          a = dizi(i)
          dizi(i) = dizi(j)
          dizi(j) = a
          a = dizino(i)
          dizino(i) = dizino(j)
          dizino(j) = a
        End If
    Next
Next
End Sub


Public Sub crossover()
Dim s(), t As Variant
Dim a As Byte
crossprob = Range("xoverprob").Value
t = 0
              ReDim s(0) As Variant
              ReDim s(1 To u) As Variant

  r = Rnd
   If r < crossprob Then
      Do
         p2 = Int(pop_sayisi * Rnd) + 1
      Loop Until (p <> p2 And ceza(p2) <= ortceza)
If adim = 1 Then
   jjson = ilkgrup - m_sayisi
Else
   jjson = u - ilkgrup
End If
      i1 = 1

       nokta1 = Int(Rnd * (jjson - 1)) + 1
       Do
          nokta2 = Int(Rnd * (jjson - nokta1)) + 1 + nokta1
       Loop Until nokta1 < nokta2
       j = 1
```

```
t = 0

For m = 1 To m_sayisi
      If j = jjson And i1 < jjson Then
         Exit For
      End If

   For k = 1 To misadet(p, m)
      ata = 0
      Do
      If j = jjson And i1 < jjson Then
         Exit For
      End If

            If i1 <= nokta2 And i1 >= nokta1 Then
               kromozom(p, i1) = kromozom(p, i1) 'ilk cocuk
               i1 = i1 + 1
         ata = 1
            Else
               For z = nokta1 To nokta2
                  If kromozom(p2, j) = kromozom(p, z) Then
                     j = j + 1
                     atla = 1
                     Exit For
                  End If
               Next
               If atla = 0 Then
      If j > jjson Or i1 < jjson Then
         Exit For
      End If
                     If Range("ISLEMSURESI").Cells(kromozom(p2, j), m) <> 100 Then
                        kromozom(p, i1) = kromozom(p2, j) 'ilk cocuk
                        Sheets("anasayfa").Cells(33, i1 + 1) = kromozom(p, i1)
                        i1 = i1 + 1
                        j = j + 1

                        ata = 1
                     Else
                        t = t + 1
                        s(t) = kromozom(p2, j)
                        j = j + 1

                     End If
               Else: atla = 0

               End If
            End If
      Loop Until ata = 1
      If j = jjson And i1 < jjson Then
         Exit For
      End If
   Next
      If j = jjson And i1 < jjson Then
         Exit For
      End If
Next
For q = 1 To t
```

```
            kromozom(p, i1) = s(q) 'ilk cocuk
                        Sheets("anasayfa").Cells(33, i1 + 1) = kromozom(p, i1)
            i1 = i1 + 1
        Next
        If nokta2 = jjson Then
            For j = nokta1 To nokta2
                kromozom(p, j) = kromozom(p, j) 'ilk cocuk
            Next

        End If
        tekhesapla
    End If
End Sub



Public Sub eniyibelirle()
Dim eniyi, enkotu As Variant
Dim eniyino, enkotuno As Integer
eniyi = ceza(1)
eniyino = 1
enkotu = 0
enkotuno = 1

For p = 2 To pop_sayisi
    If ceza(p) < eniyi Then
        eniyino = p
        eniyi = ceza(p)
    End If
    If ceza(p) > enkotu Then
        enkotuno = p
        enkotu = ceza(p)
    End If
Next
'en iyi kromozom 1 nolu kromozom yerine yaz
If adim = 1 Then
    sonis = ilkgrup - m_sayisi
Else
    sonis = u - m_sayisi
End If
For i = 1 To sonis
    kromozom(1, i) = kromozom(eniyino, i)
    bitis(1, kromozom(eniyino, i)) = bitis(eniyino, kromozom(eniyino, i))
Next
For i = 1 To m_sayisi
        misadet(1, i) = misadet(eniyino, i)
        Sheets("anasayfa").Cells(16, i + 1) = misadet(1, i)
Next

ceza(1) = ceza(eniyino)
    enyuksek(1) = enyuksek(eniyino)
    enyuksekno(1) = enyuksekno(eniyino)
    endusuk(1) = endusuk(eniyino)
    endusukno(1) = endusukno(eniyino)
topmisadet = 0
For i = 1 To m_sayisi
    Sheets("anasayfa").Cells(16, i + 1) = misadet(eniyino, i)
```

```
   topmisadet = topmisadet + misadet(eniyino, i)

        Worksheets("anasayfa").Cells(18, i + 1) = bitis(eniyino, kromozom(eniyino, topmisadet))

Next
Sheets("anasayfa").Cells(12, 2) = eniyino
Sheets("anasayfa").Cells(13, 2) = eniyi
End Sub

Public Sub mantiksalmaxyuk()
Dim a, eniyi(), eniyim(), kume() As Variant
Dim basi() As Variant
Dim eniyiobj As Variant
ReDim kume(1 To u) As Variant
ReDim islemsuresi(1 To u, 1 To m_sayisi) As Variant
ReDim hazirlik(1 To u, 1 To u) As Variant
ReDim sip(1 To u) As Variant
ReDim basi(1 To m_sayisi) As Variant
ReDim eniyi(1 To u) As Variant
ReDim eniyim(1 To m_sayisi) As Variant

If adim = 1 Then
   For s = 1 To ilkgrup
      kume(s) = s
   Next
Else
   For s = 1 To m_sayisi
      kume(s) = s
   Next
   For s = 1 To u - ilkgrup
      kume(s + m_sayisi) = ilkgrup + s
   Next


End If

If adim = 1 Then

   For i = 1 To ilkgrup
      sip(i) = Range("SIPARIS").Cells(i)
      For m = 1 To m_sayisi
         islemsuresi(i, m) = Range("ISLEMSURESI").Cells(i, m)
         For j = 1 To ilkgrup
            hazirlik(i, j) = Range("HAZIRLIK").Cells(i, j)
         Next
      Next

   Next
Else
   verihazirla

End If

If adim = 1 Then

      For m = 1 To m_sayisi
         misadet(p, m) = Int(ilkgrup / m_sayisi)
```

```
        Next
        kalan = ilkgrup Mod m_sayisi
        For b = 1 To kalan
           misadet(p, m_sayisi - b) = misadet(p, m_sayisi - b) + 1
        Next
        For m = 1 To m_sayisi
           Worksheets("anasayfa").Cells(16, m + 1) = misadet(p, m)
        Next

Else

        For m = 1 To m_sayisi
           misadet(p, m) = Int((u - ilkgrup + m_sayisi) / m_sayisi)
        Next
        kalan = (u - ilkgrup + m_sayisi) Mod m_sayisi
        For b = 1 To kalan
           misadet(p, m_sayisi - b) = misadet(p, m_sayisi - b) + 1
        Next
        For m = 1 To m_sayisi
           Worksheets("anasayfa").Cells(16, m + 1) = misadet(p, m)
        Next

End If
b = 0
If adim = 1 Then
sayi = ilkgrup
Else: sayi = u - ilkgrup + m_sayisi
End If
h = 0

For m = 1 To m_sayisi
   h = h + 1
   bas = 1
   kromozom(p, h) = kume(bas)
      Worksheets("anasayfa").Cells(15, h + 1) = kromozom(p, h)

   For ad = 1 To misadet(p, m) - 1
      Min = 1000
      For j = m_sayisi + 1 - b To sayi
            If bas <> j Then
               If hazirlik(bas, j) + islemsuresi(j, m) < Min Then
                  minno = j
                  Min = hazirlik(bas, j) + islemsuresi(j, m)
                  a = kume(minno)
               End If
            End If
      Next
      sayi = sayi - 1
      h = h + 1
      kromozom(p, h) = kume(minno)
      If ad = 1 Then
         b = b + 1
      End If
      For t = bas To sayi
         kume(t) = kume(t + 1)
      Next
         kume(sayi + 1) = 0
```

```
        For t = 1 To sayi
           For k = bas To sayi
              hazirlik(t, k) = hazirlik(t, k + 1)
           Next
        Next
        For t = bas To sayi - 1
           For k = 1 To sayi - 1
              hazirlik(t, k) = hazirlik(t + 1, k)
           Next
        Next
        For t = bas To sayi
           For mt = 1 To m_sayisi
              islemsuresi(t, mt) = islemsuresi(t + 1, mt)
           Next

        Next
        If minno > bas Then
           bas = minno - 1
        Else
           bas = minno
        End If

     Next
sayi = sayi - 1
        'kumeden baslangıc işini çıkart
        For t = bas To sayi
           kume(t) = kume(t + 1)
        Next
           kume(sayi + 1) = 0
        For t = 1 To sayi
           For k = bas To sayi - 1
              hazirlik(t, k) = hazirlik(t, k + 1)
           Next
        Next
        For t = bas To sayi - 1
           For k = 1 To sayi - 1
              hazirlik(t, k) = hazirlik(t + 1, k)
           Next
        Next
        For t = bas To sayi
           For mt = 1 To m_sayisi
              islemsuresi(t, mt) = islemsuresi(t + 1, mt)
           Next
        Next
Next
verihazirla
tekhesaplaher
If adim = 1 Then
sonis = ilkgrup
Else
sonis = u - ilkgrup + m_sayisi
End If
For i = 1 To sonis
   eniyi(i) = kromozom(p, i)
Next
If p <= 2 Then
```

```
Max = 0
For m = 1 To m_sayisi
   If Max < toplamisyuk(p, m) Then
      Max = toplamisyuk(p, m)
   End If
Next
   For i = 1 To sonis
      eniyi(i) = kromozom(p, i)
   Next
   eniyiobj = Max
   For m = 1 To m_sayisi
      eniyim(m) = misadet(p, m)
   Next

Do
iyilestirmax
tekhesaplaher
Max = 0
For m = 1 To m_sayisi
   If Max < toplamisyuk(p, m) Then
      Max = toplamisyuk(p, m)
   End If
Next
If eniyiobj > Max Then
   For i = 1 To sonis
      eniyi(i) = kromozom(p, i)
   Next
   eniyiobj = Max
   For m = 1 To m_sayisi
      eniyim(m) = misadet(p, m)
   Next
   kac = 0
Else
   kac = kac + 1
End If

Loop Until kac = 30

   For i = 1 To sonis
       kromozom(p, i) = eniyi(i)
   Next
   For m = 1 To m_sayisi
      misadet(p, m) = eniyim(m)
   Next
tekhesaplaher

j = 1
For m = 1 To m_sayisi
   For i = 1 To misadet(p, m) - 1
      kromozom(p, j) = kromozom(p, j + m)
      j = j + 1
   Next
Next
For i = 0 To m_sayisi - 1
   misadet(p, i + 1) = misadet(p, i + 1) - 1
   kromozom(p, sonis - i) = ""
Next
```

```
                p = 2
                   For i = 1 To sonis
                       kromozom(p, i) = kromozom(1, i)
                   Next
                   For m = 1 To m_sayisi
                      misadet(p, m) = misadet(1, m)
                   Next
                tekhesaplaher

If adim = 2 Then

   For p = 1 To
i = 0
j = 0
r = 0

      For m = 1 To m_sayisi

         For t = 1 To misadet(p, m_sayisi - m + 1)
            kromozom(p, u - i - m_sayisi) = kromozom(p, u - ilkgrup - j)

            i = i + 1
            j = j + 1
         Next
         For t = 1 To gecicim(1, m_sayisi - m + 1)
            kromozom(p, u - i - m_sayisi) = gecicikr(1, ilkgrup - m_sayisi - r)
            r = r + 1
            i = i + 1
         Next

      Next
      For m = 1 To m_sayisi
              misadet(p, m) = misadet(p, m) + gecicim(1, m)
      Next
   Next
End If
p = 2

Else
                For i = 1 To sonis
                   Top = Top + bitis(p, i)

                Next
                eniyiobj = Top
                For m = 1 To m_sayisi
                   eniyim(m) = misadet(p, m)
                Next

                Do
                iyilestirtop
                tekhesaplaher
                Top = 0
                For i = 1 To sonis
                   Top = Top + bitis(p, i)
                Next
                If eniyiobj > Top Then
```

```
                For i = 1 To sonis
                    eniyi(i) = kromozom(p, i)
                Next
                eniyiobj = Top
                For m = 1 To m_sayisi
                    eniyim(m) = misadet(p, m)
                Next
                kac = 0
            Else
                kac = kac + 1
            End If

            Loop Until kac = 30

                For i = 1 To sonis
                    kromozom(p, i) = eniyi(i)
                Next
                For m = 1 To m_sayisi
                    misadet(p, m) = eniyim(m)
                Next
            tekhesaplaher
            j = 1
            For m = 1 To m_sayisi
                For i = 1 To misadet(p, m) - 1
                    kromozom(p, j) = kromozom(p, j + m)
                    j = j + 1
                Next
            Next
            For i = 0 To m_sayisi - 1
                misadet(p, i + 1) = misadet(p, i + 1) - 1

                kromozom(p, sonis - i) = ""
            Next


If adim = 2 Then

            i = 0
            j = 0
            r = 0

    For m = 1 To m_sayisi
        For t = 1 To misadet(p, m_sayisi - m + 1)
            kromozom(p, u - m_sayisi - i) = kromozom(p, u - ilkgrup - j)
            i = i + 1
            j = j + 1
        Next
        For t = 1 To gecicim(1, m)
            kromozom(p, u - m_sayisi - i) = gecicikr(1, ilkgrup - m_sayisi - r)
            r = r + 1
            i = i + 1
        Next
    Next
    For m = 1 To m_sayisi
            misadet(p, m) = misadet(p, m) + gecicim(1, m)
    Next
End If
```

```
End If
End Sub


Public Sub iyilestirtop()
Dim son() As Integer
Dim ok As Byte

ReDim son(0 To m_sayisi) As Integer

maxisyukno = dizino(m_sayisi)
son(0) = 0
son(1) = misadet(p, 1)
For m = 2 To m_sayisi
   son(m) = son(m - 1) + misadet(p, m)
Next
m = 1
Do
minisyukno = dizino(m)
j = 0
fark = toplamisyuk(p, maxisyukno) - toplamisyuk(p, minisyukno)
         For t = 0 To misadet(p, maxisyukno) - 2
            aktaris = kromozom(p, son(maxisyukno) - t)
            If islemsuresi(aktaris, minisyukno) < 100 Then
               aktarisno = son(maxisyukno) - t

                     mini = 100000
                     For i = 1 To misadet(p, minisyukno)
                        If mini > hazirlik(Int(kromozom(p, son(minisyukno - 1) + i)), Int(aktaris)) Then
                           mini = hazirlik(Int(kromozom(p, son(minisyukno - 1) + i)), Int(aktaris))
                           minino = son(minisyukno - 1) + i
                        End If

                     Next
                     isilk = islemsuresi(aktaris, maxisyukno) * sip(aktaris)
                     isson = islemsuresi(aktaris, minisyukno) * sip(aktaris)
                     If aktarisno + 1 > son(maxisyukno) Then
                     hce = 0
                     hde = 0
                     Else
                        hce = hazirlik(Int(kromozom(p, aktarisno)), Int(kromozom(p, aktarisno + 1)))
                        If aktarisno - 1 < son(maxisyukno) - misadet(p, maxisyukno) + 1 Then
                           hde = 0
                        Else
                              hde = hazirlik(Int(kromozom(p, aktarisno - 1)), Int(kromozom(p, aktarisno + 1)))
                        End If
                     End If
                     If aktarisno - 1 < son(maxisyukno) - misadet(p, maxisyukno) + 1 Then
                     hdc = 0
                     Else
                        hdc = hazirlik(Int(kromozom(p, aktarisno - 1)), Int(aktaris))
                     End If
                        hac = hazirlik(Int(kromozom(p, minino)), Int(aktaris))

                     If minino + 1 > son(miniisyukno) Then
                        hcb = 0
                        hab = 0
```

```
                    Else
                    hcb = hazirlik(Int(aktaris), Int(kromozom(p, minino + 1)))
                    hab = hazirlik(Int(kromozom(p, minino)), Int(kromozom(p, minino + 1)))
                    End If
                    yenitop = -isilk + isson - hdc + hac + hcb - hab - hce + hde
                    If yenitop < 0 Then
                       f = Abs(aktarisno - minino)
                       If aktarisno > minino Then
                          'dikaaattt
                          For j = 1 To f - 1
                             kromozom(p, aktarisno - j + 1) = kromozom(p, aktarisno - j)
                          Next
                          kromozom(p, minino + 1) = aktaris
                       Else
                          For j = 1 To f
                             kromozom(p, aktarisno + j - 1) = kromozom(p, aktarisno + j)
                          Next
                          kromozom(p, minino) = Int(aktaris)
                       End If
                       misadet(p, maxisyukno) = misadet(p, maxisyukno) - 1
                       misadet(p, minisyukno) = misadet(p, minisyukno) + 1

                       ok = 1

                    End If
                 Exit For

              End If
           Next

           m = m + 1
Loop Until (ok = 1 Or m = m_sayisi)
End Sub

Public Sub iyilestirmax()
Dim son() As Integer
Dim ok As Byte

ReDim son(0 To m_sayisi) As Integer

maxisyukno = dizino(m_sayisi)
son(0) = 0
son(1) = misadet(p, 1)
For m = 2 To m_sayisi
   son(m) = son(m - 1) + misadet(p, m)
Next
m = 1
Do
minisyukno = dizino(m)
j = 0
fark = toplamisyuk(p, maxisyukno) - toplamisyuk(p, minisyukno)
           For t = 0 To misadet(p, maxisyukno) - 2
              aktaris = kromozom(p, son(maxisyukno) - t)
              If islemsuresi(aktaris, minisyukno) < 100 Then
                 aktarisno = son(maxisyukno) - t

                    mini = 100000
```

```
For i = 1 To misadet(p, minisyukno)
   If mini > hazirlik(Int(kromozom(p, son(minisyukno - 1) + i)), Int(aktaris)) Then
      mini = hazirlik(Int(kromozom(p, son(minisyukno - 1) + i)), Int(aktaris))
      minino = son(minisyukno - 1) + i
   End If

Next
isilk = islemsuresi(aktaris, maxisyukno) * sip(aktaris)
isson = islemsuresi(aktaris, minisyukno) * sip(aktaris)
If aktarisno + 1 > son(maxisyukno) Then
hce = 0
hde = 0
Else
   hce = hazirlik(Int(kromozom(p, aktarisno)), Int(kromozom(p, aktarisno + 1)))
   If aktarisno - 1 < son(maxisyukno) - misadet(p, maxisyukno) + 1 Then
      hde = 0
   Else
         hde = hazirlik(Int(kromozom(p, aktarisno - 1)), Int(kromozom(p, aktarisno + 1)))
   End If
End If
If aktarisno - 1 < son(maxisyukno) - misadet(p, maxisyukno) + 1 Then
hdc = 0
Else
   hdc = hazirlik(Int(kromozom(p, aktarisno - 1)), Int(aktaris))
End If
   hac = hazirlik(Int(kromozom(p, minino)), Int(aktaris))

If minino + 1 > son(minisyukno) Then
   hcb = 0
   hab = 0
Else
hcb = hazirlik(Int(aktaris), Int(kromozom(p, minino + 1)))
hab = hazirlik(Int(kromozom(p, minino)), Int(kromozom(p, minino + 1)))
End If
yminisyuk = toplamisyuk(p, minisyukno) + hac + hcb + isson - hab
ymaxisyuk = toplamisyuk(p, maxisyukno) - isilk - hdc + hde - hce
If yminisyuk > ymaxisyuk Then
   yenimax = yminisyuk
Else: yenimax = ymaxisyuk
End If

If toplamisyuk(p, minisyukno) > toplamisyuk(p, maxisyukno) Then
   eskimax = toplamisyuk(p, minisyukno)
Else: eskimax = toplamisyuk(p, maxisyukno)
End If
If yenimax < eskimax Then
   f = Abs(aktarisno - minino)
   If aktarisno > minino Then
      For j = 1 To f - 1
         kromozom(p, aktarisno - j + 1) = kromozom(p, aktarisno - j)
      Next
      kromozom(p, minino + 1) = aktaris
   Else
      For j = 1 To f
         kromozom(p, aktarisno + j - 1) = kromozom(p, aktarisno + j)
      Next
      kromozom(p, minino) = Int(aktaris)
```

```
                End If
                misadet(p, maxisyukno) = misadet(p, maxisyukno) - 1
                misadet(p, minisyukno) = misadet(p, minisyukno) + 1
                ok = 1
            End If
        Exit For

      End If
    Next

    m = m + 1
Loop Until (ok = 1 Or m = m_sayisi)
End Sub


Public Sub verihazirla()
If adim = 2 Then

For i = 1 To m_sayisi
  For j = 1 To m_sayisi
    hazirlik(i, j) = Range("HAZIRLIK").Cells(Int(ilkatama(i)), Int(ilkatama(j)))

  Next
  For t = 1 To u - ilkgrup
    hazirlik(i, m_sayisi + t) = Range("HAZIRLIK").Cells(Int(ilkatama(i)), ilkgrup + t)

  Next
Next

For i = 1 To u - ilkgrup
  For t = 1 To m_sayisi
    hazirlik(m_sayisi + i, t) = Range("HAZIRLIK").Cells(ilkgrup + i, Int(ilkatama(t)))
  Next
  For t = 1 To u - ilkgrup
    hazirlik(m_sayisi + i, m_sayisi + t) = Range("HAZIRLIK").Cells(ilkgrup + i, ilkgrup + t)
+ i, m_sayisi + t)

  Next
Next
  For i = 1 To m_sayisi
    For m = 1 To m_sayisi

    islemsuresi(i, m) = Range("ISLEMSURESI").Cells(Int(ilkatama(i)), m)


    Next
  Next
  For i = 1 To u - ilkgrup
    For m = 1 To m_sayisi
      islemsuresi(m_sayisi + i, m) = Range("ISLEMSURESI").Cells(ilkgrup + i, m)
    Next
  Next
  For i = 1 To m_sayisi
    sip(i) = Range("SIPARIS").Cells(Int(ilkatama(i)))
  Next
  For i = 1 To u - ilkgrup
    sip(m_sayisi + i) = Range("SIPARIS").Cells(ilkgrup + i)
```

```
   Next
Else

For i = 1 To ilkgrup
      sip(i) = Range("SIPARIS").Cells(i)
      For m = 1 To m_sayisi
         islemsuresi(i, m) = Range("ISLEMSURESI").Cells(i, m)
      Next
      For j = 1 To ilkgrup
      hazirlik(i, j) = Range("HAZIRLIK").Cells(i, j)
      Next
Next
End If
End Sub

Public Sub sabitle()
If adim = 1 Then
   sonis = ilkgrup
Else
   sonis = u - ilkgrup + m_sayisi
End If
For i = 1 To sonis - m_sayisi
   gecicikr(adim, i) = kromozom(1, i)
Next
For m = 1 To m_sayisi
   gecicim(adim, m) = misadet(1, m)
Next
If adim = 1 Then
   For m = 1 To m_sayisi
      f = f + misadet(1, m)
      ilkatama(m) = kromozom(1, f)
   Next
End If
End Sub
```