TESİS YERLEŞİM PROBLEMLERİ İÇİN TAKIM ZEKASI TABANLI BİR RASSAL ENİYİLEME ALGORİTMASI

Fehime UTKAN

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Temmuz 2006

A SWARM INTELLIGENCE BASED STOCHASTIC OPTIMIZATION ALGORITHM FOR FACILITY LAYOUT PROBLEMS

Fehime UTKAN

MASTER OF SCIENCE THESIS

Department of Industrial Engineering

July 2006

TESİS YERLEŞİM PROBLEMLERİ İÇİN TAKIM ZEKASI TABANLI BİR RASSAL ENİYİLEME ALGORİTMASI

Fehime UTKAN

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

Danışman: Yrd. Doç.Dr. Muzaffer KAPANOĞLU

Temmuz 2006

Fehime UTKAN' ın YÜKSEK LİSANS tezi olarak hazırladığı "**Tesis Yerleşim Problemleri için Takım Zekası Tabanlı Bir Rassal Eniyileme Algoritması**" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Üye :

Üye :

Üye :

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

Prof. Dr. Abdurrahman KARAMANCIOĞLU

Enstitü Müdürü

İÇİNDEKİLER DİZİNİ

ÖZET vii
ABSTRACT viii
TEŞEKKÜR ix
ŞEKİLLER DİZİNİ x
TABLOLAR DİZİNİ xi
SİMGELER ve KISALTMALAR DİZİNİ xii

1. GİRİŞ 1

2. TESİS YERLEŞİM DÜZENLEMESİ 4
2.1. Yerleşim Düzenlemesinin Önemi 5
2.2. Yerleşim Düzenlemesinin Amacı ve Kapsamı 6
2.3. Tesis Yerleşim Şekilleri 7
2.4. Yeniden Yerleşimde Dikkate Alınması Gereken Hususlar 10
2.5. Tesis Yerleşimi ile İlgili Çalışmalar ve Yeni Eğilimler 12
2.6. Bilgisayar Destekli Tesis Planlaması 15
2.6.1. Bilgisayar destekli tesis planlama algoritmalarının sınıflandırılması 15
2.6.2. Yerleşim algoritmalarında kullanılan modelleme teknik ve yöntemleri 18

3. KARELİ ATAMA PROBLEMLERİ 22
3.1. KAP Tabanlı Tesis Yerleşimi Problemleriyle İlgili Çalışmalar 22
3.2. Kareli Atama Problemlerinin Modellenmesi 23
3.2.1. Tamsayılı doğrusal programlama 23
3.2.2. Karışık tamsayılı doğrusal programlama 25
3.3. Kareli Atama Problemlerinin Karmaşıklığı 26
3.4. KAP için Çözüm Yöntemleri 26
3.4.1. Eniyileme algoritmaları 27
3.4.2. Sezgisel algoritmalar 27
3.4.3. Üstsezgiseller (Metaheuristics) 28

4. TAKIM ZEKASI TABANLI RASSAL ENİYİLEME (TZRE) ALGORİTMASI 31
4.1. TZRE'nin Ortaya Çıkışı 31
4.2. TZRE'nin Temel İlkeleri 32
4.3. Parametre Seçimi ve Yakınsama Analizi 38
4.4. TZRE'nin Uygulama Alanları ve Yapılmış Çalışmalar 38

5. KAP TABANLI TESİS YERLEŞİM PROBLEMLERİ İÇİN TAKIM ZEKASI TABANLI BİR RASSAL ENİYİLEME ALGORİTMASI 41
5.1. Çalışmanın Amacı 41
5.2. Algoritmanın Tanıtımı 42
5.3. Kontrol Parametrelerinin Seçimi 48
5.4. Deneysel Sonuçlar 49
5.4.1. İlk problem seti deneyleri ve sonuçları 50
5.4.2. QAPLIB problemleri için yapılan deneysel çalışma ve sonuçları 51
5.4.3. Parametre Analizi 57
5.5. Bölüme Özel Kısıtları Olan KAP Tabanlı Yerleşim Problemleri 57

SONUÇ ve ÖNERİLER 62
KAYNAKLAR DİZİNİ 64
EKLER 69

ÖZET

Tesis yerleşim problemleri için ürün çeşitliliği ve taleplerindeki değişiklikler karşısında, yerleşimin yeniden değerlendirilmesini sağlayacak araçların geliştirilmesi, esnek ve modüler üretim teknolojilerinin hızla gelişmesiyle birlikte önem kazanmıştır.

Kareli atama problemleri (KAP), tesis yerleşimi problemlerinin ifade edilmesinde uzun yıllardır kullanılmaktadır. Ancak bu yaklaşımla; 15 ve daha fazla bölümlü problemler için en iyi çözümü garanti etmek zor olduğu gibi, bir çok gerçek uygulamanın gereksinimleri de karşılanamamaktadır.

Takım zekası tabanlı rassal eniyileme (TZRE) algoritması, yakın bir zamandan itibaren; özellikle sürekli problemlerin dahil olduğu bazı problem alanları için rassal ve arama tabanlı eniyileme tekniği olarak kullanılmaya başlanmasına rağmen literatürde, kesikli problemlerle ilgili yeterli sayıda kapsamlı çalışma mevcut değildir.

Bu çalışmada, kalitesi yüksek (en iyi ya da en iyiye yakın) çözümler bulmak amacıyla takım zekası tabanlı rassal bir algoritma geliştirilerek (i) bilinen KAP-tabanlı blok yerleşim problemlerine benzeyen ya da benzemeyen birtakım kısıtları olan bölümlerin atamasının yapılmasına olanak veren tesis yerleşimine (ii) rassal arama yaklaşımları alanına katkıda bulunacak yeni bir yaklaşım önerilmektedir.

Önerilen takım zekası tabanlı üç rassal eniyileme algoritması, 80'in üzerinde probleme uygulanmıştır. Ayrıca, belli bölümlerin belli yerleşim alanlarına istenmeyen atamaların yapılmaması ve bölüm boyutlarından kaynaklanan birtakım kısıtların gözönüne alınması için QAPLIB problemleri bölüm-bazlı kısıtları temsil edecek şekilde geliştirilmiştir.

Bu çalışma, TZRE'nin, KAP olarak modellenen tesis yerleşim problemlerinin çözümünde kullanılabilir bir yöntem olabileceğini göstermektedir.

Anahtar Kelimeler: Tesis Yerleşim Problemleri, Kareli Atama Problemleri, Takım Zekası

ABSTRACT

Development of new tools enables the evaluation of facility layouts against the changes of product mix and demand fluctuation have gained importance with advances in flexible and modular manufacturing technologies.

The quadratic assignment problem (QAP) has been used to deal with facility layout problems extensively for decades. This approach can neither resolve the intractability of larger problems (with more than 15 departments) by assuming the optimal solutions nor satisfy the requirements of many real-world applications.

Swarm intelligence based stochastic optimization (SISO) algorithm has been recently introduced as a stochastic, search-based optimization method for numerous problem domains especially including continous problems. There haven't been enough researches about discrete problems yet.

In this study, a new approach that will contribute to (i) the facility layout domain by enabling the assignment of departments with some restrictions similar and unsimilar traditional QAP-based block layout problems, and (ii) the stochastic search methods domain by developing a swarm intelligence based stochastic optimization algorithm is proposed to find high-quality (optimal or near-optimal) solutions efficiently in a robust manner.

Proposed SISO approach is tested on more than 80 problem instances. In order to represent some department-specific restrictions for generating more practical test problems including undesired assignment of certain departments to certain locations as well some considerations related to department sizes.

This study reveals that SISO is a promising algorithm for solving plant layout problems formulated as quadratic assignment problems.

Keywords: Facility Layout Problems, Quadratic Assignment Problems, Swarm Intelligence, Particle Swarm Optimization

TEŞEKKÜR

Bu çalışmayı gerçekleştirmemde çok büyük katkısı olan, akademik alanda her yönüyle örnek aldığım değerli danışmanım Yrd. Doç. Dr. Muzaffer KAPANOĞLU'na verdiği destek ve gösterdiği sabır için teşekkürlerimi bir borç bilirim.

Araş. Gör. Mete ALİKALFA'ya programlama konusundaki desteği; Onur KURT'a deneysel çalışmalarda verdiği destek için teşekkür ederim.

Yaşamım boyunca her anlamda yanımda olan aileme, destekleri ve bana kattıkları herşey için teşekkürlerimi bir borç bilirim.

ŞEKİLLER DİZİNİ

Şekil 2.1 Blok yerleşim ve ayrıntılı yerleşim 7

Şekil 2.2 Kesikli yerleşim gösterimi 17

Şekil 2.3 Sürekli yerleşim gösterimi 17

Şekil 2.4 Boşluk dolduran eğriler 20

Şekil 4.1 Sürü içindeki bireyin hareket bileşenleri 32

Şekil 4.2 Çözüm alanında bireylerin arama hareketleri 36

Şekil 4.3 TZRE'nin genel akış şeması 37

Şekil 4.4 Yıllar bazında yayımlanmış çalışma sayıları 39

Şekil 5.1 Boyutları eşit olan/olmayan iş istasyonları ve bölümlerin yerleşim noktalarına atanmasının gösterimi 41

Şekil 5.2 İş istasyonlarının TZRE ile yerleşim noktalarına atanmasının gösterimi 42

Şekil 5.3 Birey gösterimi 43

Şekil 5.4 Takım Zekası Tabanlı Rassal Eniyileme Algoritması (TZRE) adımları 45

Şekil 5.5 Taslak TZRE algoritması 46

Şekil 5.6 Rassal anahtar kodlaması 47

Şekil 5.7 Önerilen TZRE algoritması için geliştirilen program arayüzü 49

Şekil 5.8 Problem büyüklüklerine göre TZRE algoritmalarının karşılaştırılması 55

Şekil 5.9 TZRE algoritmalarının yakınsama yüzdelerinin karşılaştırılması 56

Şekil 5.10 Örnek Lingo sonuç ekranı 60

TABLOLAR DİZİNİ

Tablo 2-1 Tesis Yerleşimi Çalışmalarının Sınıflandırması 14

Tablo 5-1 TZRE algoritmasının ilk deneysel sonuçları 50

Tablo 5-2 QAPLIB problemleri için deneysel sonuçlar 52

Tablo 5-3 QAPLIB problemleri için özet sonuçlar 53

Tablo 5-4 Problem büyüklükleriyle yakınsama oranlarının karşılaştırılması 54

Tablo 5-5 Farklı parametre değerleri için deneysel sonuçlar 58

Tablo 5-6 Bölüm kısıtı olan problemler için deneysel sonuçlar 61

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
KAP (A,B)	A ve B katsayı matrisleri ile tanımlanan Koopmans Beckmann Kareli Atama Problemi

Kısaltmalar	Açıklama
KAP	Kareli Atama Problemi
k.a.	kısıtları altında
KTDP	Karışık Tamsayılı Doğrusal Programlama
DTYP	Dinamik Tesis Yerleşim Problemi
TZRE	Takım Zekası Tabanlı Rassal Eniyileme
STYP	Statik Tesis Yerleşim Problemi
vd.	ve diğerler

1. GİRİŞ

Tesis yerleşim problemleri, global pazardaki değişkenlik karşısında tedarik zinciri uygulamalarının hız kazanmasıyla birlikte, esnek ve modüler imalat teknolojilerinde gerçekleşen ilerlemeler nedeniyle bütünüyle yeni bir bakış açısı kazanmıştır. Günümüzün yığın-seri imalat sistemleriyle birlikte, seri üretimde birtakım uyarlamalar yapılmasının gerekliliği ortaya çıkmıştır. Yapılan uyarlamalar, ürünlerin yaşam çevrim sürelerinin kısalmasına neden olarak işletmeleri; daha sık güncellenmesi gereken ürün çeşitliliğini arttırmak zorunda bırakmaktadır. Ortaya çıkan bu yeni durumda, üretim maliyetlerinin azaltılması ve tedarik sürelerinin kısaltılabilmesi için mevcut yerleşimin doğrulanması bir ihtiyaç haline gelmiştir. Bu doğrulama süreci, mevcut yerleşimin bazen tümüyle değiştirilmesine neden olabilmektedir Bu değişiklikler, bu çalışmada "yeniden yerleşim problemleri" olarak anılacaktır. Bu nedenle, yerleşim etkinliğini değerlendirmek için kullanılabilecek anlık ya da dönemsel yaklaşımlar üzerinde durulması gereken çalışma alanı olarak, önemini korumaya devam etmektedir. Ancak, birçok kombinatoryal eniyileme probleminde olduğu gibi, sadece küçük boyutlu tesis yerleşim problemlerinin en iyi çözümü elde edilebilmektedir. Bu nedenle, tesis yerleşimi problemlerinin kareli atama problemleri olarak modellenmesi, üst-sezgisel (metaheuristic) yaklaşımları kullanan araştırmacıların ilgi odağı haline gelmiştir. Bu konuda yapılan çalışmalar incelendiğinde, karınca kolonisi optimizasyonu (Kendall Jr. ve Shang, 2006; Solimanpur vd., 2005), evrimsel algoritmalar (Dunker vd., 2005), genetik algoritmalar (İşlier, 1998; Balakrishnan ve Cheng, 2000; Balakrishnan vd., 2003; Wang vd., 2005) ve tavlama benzetimini (Baykasoglu ve Gindy, 2001; Chiang ve Chiang, 1998), içeren onlarca çalışma bulmak mümkündür.

Bu çalışmada, kareli atama problemi (quadratic assignment problem) yapısında modellenmiş; bölüme özel kısıtları olan/olmayan tesis yerleşim problemleri için geliştirilmiş Takım Zekası Tabanlı Rassal Bir Eniyileme (TZRE) algoritması önerilmektedir.

KAP, bir tesis yerleşimi problemini modellemek için 1957'de Koopmans ve Backman tarafından geliştirilmiş ve birçok araştırmaya konu olmuştur. Tesis yerleşim

problemlerini çözmek için KAP dışında diğer modeller kullanılsa da KAP, en çok kullanılan model olarak yaygınlığını korumaya devam etmektedir (Loiola vd., 2005). Diğer modeller de KAP'ın farklı türleri olarak geliştirilmiş ya da KAP'ı iskelet olarak kullanmışlardır (Chiang ve Chiang, 1998).

Önerilen TZRE algoritması tabanlı yaklaşım; kuş ve balık sürülerinin sosyal davranışlarından esinlenerek geliştirilmiş; parçacıkların birbirleriyle işbirliği yaptıkları, populasyon tabanlı bir arama gerçekleştirmektedir (Kennedy vd., 2001). TZRE, keşfetme (exploration) ve yararlanmayı (exploitation) denglemek amacıyla yerel arama teknikleri (parçacıkların bireysel deneyimleri aracılığıyla) ile bütünsel arama tekniklerini (parçacıkların komşuluk deneyimleri aracılığıyla) birleştirir. TZRE, önceki bireysel ve sosyal (sürü) deneyimlerini hatırlayarak bunları yeni nesilleri türetmede kullanır (Salman vd., 2002). Yeni bireyler için kullanılan bu türetme fikri, TZRE'yu diğer populasyon tabanlı algoritmalardan ayırmaktadır (Penev ve Littlefair, 2004). Son yıllarda TZRE, birçok araştırma ve uygulama alanlarında kullanılmakta ve diğer tekniklere göre daha yüsek başarı göstermeketdir (Hu, 2006).

Bilinen tesis planlama ve yerleşim algoritmalarının problemi uzun vadeli ve statik bir bakışla değerlendirmeleri, kombinatoryal problemlerde henüz en iyi çözümü kabul edilebilir bir sürede garanti edecek çözüm tekniklerinin olmaması ve sosyal zeka (swarm intelligence) yaklaşımının gelişen ve dikkat çeken bir zeki üretim tekniği olarak kullanılabilir olması bu çalışmayı önemli kılan hususların başında gelmektedirler. Bu çerçeve kapsamında çalışma 5 bölümden oluşmuştur.

İkinci bölümde, tesis yerleşim düzenlemesinin önemi, amaç ve kapsamı ile bilgisayar destekli tesis planlama ve tesis yerleşim şekilleri yanında mevcut çözüm yöntemlerine değinilmiştir. Bir sonraki bölümde, kareli atama problemlerinin modellenmesi, çözüm yöntemleri ve özellikle kullanılan üstsezgiseller (metaheurusitcs) ele alınmıştır. Dördüncü bölümde, bu çalışmanın odak noktasını oluşturan takım zekası tabanlı rassal algoritmanın tanımı, adımları, parametre seçimi ve uygulama alanları ile ilgili bilgi verilmiştir. Beşinci bölümde, KAP tabanlı tesis yerleşim problemleri için önerilen TZRE algoritması ayrıntılı olarak açıklanarak algoritmanın başarısını ortaya çıkartmak için yapılan deneysel çalışmalara geniş olarak yer verilmiştir.

Sonuç ve Öneriler bölümünde yapılan çalışmanın sonucu ve bu konuda ileride yapılabilecek çalışmalara değinilmiştir.

2. TESİS YERLEŞİM DÜZENLEMESİ

İşletmeler, sahip oldukları kaynaklardan en etkin şekilde yararlanmayı amaçlarlar. Bu da ancak verimli bir imalat ya da hizmet sistemine sahip olmakla mümkündür.

Bir üretim sistemi içindeki malzeme, makine ve teçhizat, işgücü gibi unsurların yerleşiminin en iyi biçimde düzenlenmesi ve en az maliyetli çözümler aranması olarak tanımlanan tesis içi yerleşim düzenlemesinin geniş ölçüde işletmelerde uygulanması 1930'lu yıllara dayanmaktadır. Ancak, geçen yüzyılda sanayi toplumuna, günümüzde de sanayi ötesi topluma geçişin getirdiği büyük ve ani değişimler, yerleşim problemlerini gelenekler ve alışkanlıklarla çözülemeyecek hale sokmuştur. Yerleşim problemleri, bireylerin tek başlarına bütünüyle kavrayıp çözemeyecekleri kadar karmaşıktır. Bu nedenle yerleşim problemlerine sistematik olarak yaklaşmak ve bu alanda oluşan bilgi birikiminden yararlanmak gerekmektedir. Yöneylem Araştırması, Yönetim Bilimi, Uygulamalı Matematik ve Bilgisayar Teknolojisi sahasındaki gelişmeler yerleşim düzenlemesi problemlerine daha köklü çözümler bulma yolunu açmış, yerleşim düzenlemesinde kullanılacak etkili kavram ve teknikler geliştirmiştir (İşlier, 1997).

Tesis yerleşim problemleri, bölümler arasındaki planlanmış etkileşimler ile ilgili olan maliyetleri en küçüklemek amacıyla; boyutları bilinen düzlemsel bir bölgenin alanları bilinen bölümlere ayrılmasını içeren bir en iyileme problemleri ailesidir. İlgili maliyetler; malzeme aktarma sisteminin kurulmasını içeren taşıma maliyetleri ya da bölümlerin yakınlık tercihlerini yansıtabilirler (Tate ve Smith, 1995). Statik Tesis Yerleşim Problemleri (STYP), çeşitli yerleşim bölgelerine farklı bölümlerin atanması ile ilgili toplam malzeme aktarma maliyetlerini en küçükler (Rosenblatt, 1986). Her birinin kendi tek akış matrisi girdisi ve blok diyagram çıktısı olan bir dizi tek dönem ya da statik problemler vardır. Amaç fonksiyonuna ilave edilen "yeniden düzenleme maliyeti (rearrangement cost)" bu statik problemleri biraraya getirir. Dinamik Tesis Yerleşim Problemi (DTYP), statik yerleşim problemlerindeki akış maliyetine ilave

olarak yeniden yerleşim maliyetini de en küçüklemeyi amaçlar (Lacksonnen ve Enscore, 1993).

Dinamik tesis yerleşimi problemi, bir tesisin içinde yeralan birimler arasındaki bilgi, belge, malzeme, insan, taşıma aracı ve ürün akışlarının zaman içinde değiştiği yerleşim problemlerini ifade eder. Bu problemlerde amaç her planlama dönemi için toplam akış maliyetleri ile yeniden yerleşim maliyetlerini gözönüne alarak yerleşim/yeniden-yerleşim kararlarını bir çok teknik kısıt altında ve birden fazla başarı ölçütünü gözönüne alarak vermektir. Problem özellikle modüler imalat teknolojilerinin yaygınlaşmaya başlaması ile dikkat çekmeye başlamış olup çözümü güç problemler arasında gösterilmektedir (Erel vd., 2003).

2.1. Yerleşim Düzenlemesinin Önemi

Amerika Birleşik Devletleri'nde kişi başına düşen milli gelirin yaklaşık yüzde sekizi, 1955 yılından beri, her yıl yeni tesislerin yapımına harcanmaktadır (Rosenblatt, 1986). Buna ek olarak, mevcut tesislerin büyük bir kısmında artan performans gereksinimlerinin karşılanabilmesi için sürekli olarak yeniden yerleşim çalışmaları yapılmaktadır. Bu veriler, yılda iki yüz elli milyar Amerikan Doları'nın tümleşik planlama sürecinin bileşenleri olan; tesis sistemleri, yerleşimi, taşıma sistemleri ve tesis yer seçimine harcandığını göstermektedir (Tompkins, 1997).

Genel tesis yerleşimi problemleri, bir tesis içindeki, birbirleriyle eşit olmayan alan gereksinimlerine sahip bölümlerin, üst üste gelmeyecek şekildeki en *"etkin"* düzenlenmesini bulmaya yöneliktir.

Tesis yerleşim problemlerinin çözümünün; malzeme taşıma maliyeti, tesis içindeki boş alanların kullanımı, bölümlerin şekilleri, tesisin genişletilebilmesinde kolaylık sağlaması, esneklik, ısıtma/soğutma gereksinimleri gibi birçok konu üzerinde çarpıcı önemi olmasına rağmen, *"etkin"* olmanın herkes tarafından kabul görmüş bir tanımı yoktur. Bununla birlikte, yerleşim çözümleri genellikle malzeme taşıma maliyetlerini en küçüklemek amacıyla geliştirilirler.

Tesis yerleşimi ve malzeme taşıma sistemleri konusunun önemi, Tompkins ve White (1984) tarafından izleyen şekilde belirtilmiştir: "İmalattaki toplam işleyiş maliyetinin %20 - %50 sini malzeme taşıma maliyetleri oluşturmaktadır ve *etkin* bir yerleşim bu maliyetleri en azından %10 -%30 oranında azaltabilir (Meller ve Gau, 1996). Bu iddia, Nicol ve Hollier (1983) tarafından Büyük Britanya'da 33 adet şirket için yapılan araştırma ile de desteklenmiştir. Ayrica, Nicole ve Hollier, toplam işçilik maliyetinin %12'sinin, tesis yerleşimine doğrudan bağlı olan taşıma, depolama ve ulaşımla ilgili görevleri kapsadığını belirtmişlerdir (Rosenblatt, 1986).

2.2. Yerleşim Düzenlemesinin Amacı ve Kapsamı

Tesis yerleşiminin en temel amacı, öngörülen kapasite ve kalite gereklerini en ekonomik biçimde karşılayan bir üretim veya hizmet sistemi geliştirmektir. Bu anlamda tesis yerleşimi, mal veya hizmet üretimine yönelik faaliyetlere ilişkin canlı ve cansız varlıkların tümünün hareket miktarlarını en az düzeye indirmeyi amaçlar. Bu çerçeveye göre tesis yerleşiminin amaçları aşağıda maddeler halinde verilmiştir:

1. Malzeme ve personel hareketlerini en aza indirecek ve hareketleri etkili biçimde kontrol edecek bir mekanizma kurarak üretim sürecinin kolaylaştırılması
2. Malzeme yükleme, boşaltma ve taşımaların en aza indirilmesi
3. Yarı-mamullerin devir hızının yükseltilmesi
4. Mümkün gelişmeleri karşılayacak ölçüde esneklik sağlanması
5. Malzeme, iş gücü, donanım ve yer gibi üretim faktörlerinin daha etkin kullanılması
6. Sabit yatırımların en alt düzeyde tutulması
7. Denetim ve yönetimde etkinlik
8. Çalışma koşullarının iyileştirilmesi

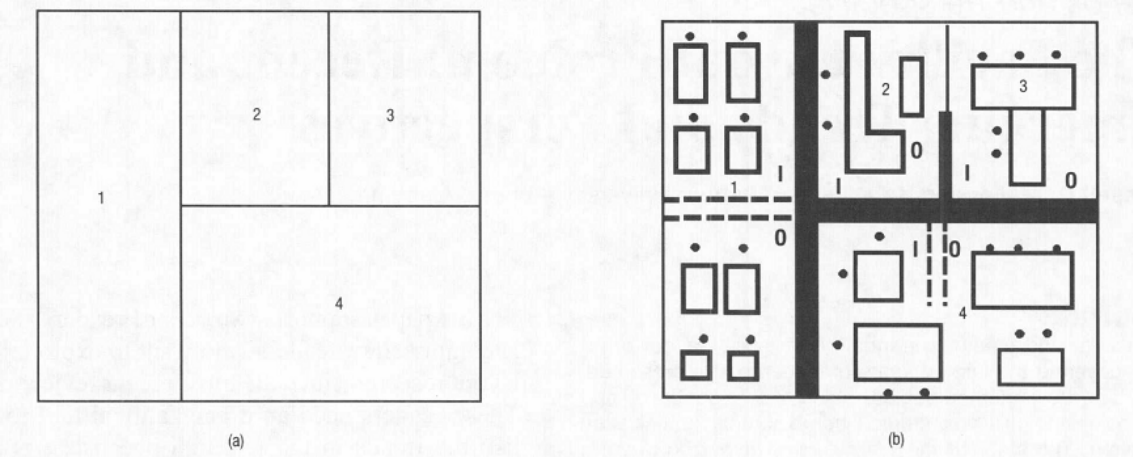
Bu amaçların tümünü de aynı ölçüde gerçekleştirmek çok güçtür. Örneğin, taşımaların en aza indirilmesi, sürece göre düzenlenmesi gereken bir işyeri için en önemli amaç olurken, ürüne göre düzenlemede, bu önemini kaybetmekte; bunun yerine, üretim hattının dengelenmesi ön plana çıkmaktadır. Öte yandan, malzeme, işgücü,

donanım ve yer gibi faktörlerin etkin olarak kullanılması amacı, mümkün gelişmeleri karşılayacak esnekliği önemli ölçüde azaltmaktadır.

İdeal bir üretim sistemi, uygulanması düşünülen çeşitli üretim programlarına gereken miktarda donanım, çalışma ve depolama alanı sağlayabilecek şekilde tasarlanmalıdır. Bu sistem, ayrıca, malzeme, parça, yarı-mamul ve mamullerin sistem içinde hareketlerini sağlayan bir taşıma sistemi; bakım ve onarım için takımhaneler, tamir-bakım atölyeleri; personeli için de, sağlık, beslenme ve eğitime yönelik sosyal tesisler içermelidir. Tüm bunların tek bir tesis bünyesinde konumlandırılması ve koordinasyonu, işyeri düzenlemenin kapsamını oluşturmaktadır (Su ve Aslan, 1997).

2.3. Tesis Yerleşim Şekilleri

Tesis yerleşim problemlerinin çıktısı, herbir bölümün göreceli yerleşimini gösteren blok yerleşimidir (Şekil 2.1a). Tam olarak bölüm yerleşimlerini, koridor yapılarını ve herbir bölümün iç yerleşimini gösteren ayrıntılı yerleşim de bir sonraki adım olarak elde edilebilir (Şekil 2.1b) (Meller ve Gau, 1996).



Şekil 2.1 Blok yerleşim ve ayrıntılı yerleşim

Tesis yerleşim şekilleri ve birbirlerine göre üstün ve sakıncalı yanları izleyen sayfada verilmiştir (Tompkins vd., 2003):

a. Ürün Akışına Göre Yerleşim

Ürün akışına göre yerleşimde, ürün rotası esas alınır. Rota imal edilen parçaların atölye içinde izlediği yoldur. Akış esaslı yerleşim düzenlenmesinde, rotaların bir hat boyunca sürmesi hedeflenir. Sabit otomasyon ve özellikle iletim hatları bu tür yerleşime dayanmaktadır.

- **Ürün Akışına Göre Yerleşimin Üstünlükleri**
    - Basit, düzgün, anlamlı akış hatları oluşur
    - Ara stoklar çok azalır
    - Birim üretim süreleri düşer
    - Daha az malzeme aktarımı olur
    - Özel amaçlı basit makineler de kullanılabilir
    - Akış süreleri kısadır
    - Daha sade bir üretim planlama ve kontrol sistemi yeterli olabilir
    - Niteliksiz iş gücü kullanılabilir

- **Ürün Akışına Göre Yerleşimin Sakıncaları**
    - Süreç esnekliğinden yoksundur
    - Zamanlamada esneklik yoktur
    - Büyük yatırım gerektirir (özel amaçlı donanım)
    - Bir tezgahın arızası, hattın bütünüyle durmasını sebep olur
    - Üretim hızını en yavaş olan tezgah belirler
    - Ürün tasarımında değişiklik, hattı kullanılmaz duruma getirir
    - İşçiler için monotonluk demektir

b. Sürece Göre Yerleşim

Bu yerleşim tipinde benzer işi yapan makineler bir araya getirilir.

- **Sürece Göre Yerleşimin Üstünlükleri**
    - Donanım ve işgücünün atanması esnektir
    - Genel amaçlı tezgahlar kullanılabilir
    - Tezgah kullanım oranları daha yüksektir

- Daha az yatırım yapılır (tekrarlanan tezgah yok)
- İşçi ve amirinde uzmanlaşma sağlanır

- **Sürece Göre Yerleşimin Sakıncaları**
  - Monotonluk daha azdır
  - Malzeme aktarma verimsizdir
  - Büyük ara stoklar oluşur
  - Boş beklemeler çoktur
  - Üretim planlaması ve kontrolü karmaşıktır
  - Nitelikli işgücü gerektirir
  - Ayar süreleri ve öğrenme süreci, verimi düşürür

c. Sabit Konumlu Ürüne Göre Yerleşim

Sabit konumlu ürüne göre yerleşimde gemi,uçak gibi taşınması güç,büyük ve ağır ürünler sabit bir konumda dururlar, gerektiğinde işçiler ve makineler bu ürünün ilgili noktalarına gelirler.

- **Sabit Konumlu Ürüne Göre Yerleşimin Üstünlükleri**
  - Malzeme hareketleri azalır
  - Takım çalışması ile süreklilik, sorumluluk ve dolayısıyla kalite bilinci sağlanır
  - İş zenginleştirme fırsatları doğar
  - Ürün tasarımı, ürün karışımı ve üretim hacmindeki değişikliklere karşı, oldukça esnektir
  - Parça bölümden bölüme dolaşmadığından yeniden eğitim ve görevlendirme problemleri yoktur
  - Taşıma maliyetleri ve hasar tehlikesi azalır

- **Sabit Konumlu Ürüne Göre Yerleşimin Sakıncaları**
  - Yetenekli ve esnek işçiler gerektirir
  - İnsan ve makinaların araziye taşınması pahalı olabilir
  - Donanımdan yararlanma oranı düşük olabilir

- Aynı donanımdan birkaç adet gerekebilir
- Daha geniş alan ve daha fazla yarımamul gerekebilir
- Üretimin çizelgelenmesi ve kontrolunda daha çok eşgüdüm gerekebilir

d. Hücresel Üretime Göre YerIeşim

Bu üretim tipinde ,aynı işlemleri gören parçalar, aynı hücrelerde üretilmektedir. Böyle bir üretim hücresinde, o parça ailesini üreten birkaç tezgah, o tezgahları kullanan birkaç işçi yer almaktadır.

- **Hücresel Üretime Göre YerIeşimin Üstünlükleri**
  - Ürüne ve sürece göre yerleşimin iyi yönlerini kendinde birleştirmiştir
  - Ürünlerin gruplanmasıyla, tezgah kullanım oranları yükseltilebilir.
  - Oldukça düzgün akışlar ve kısa taşıma mesafeleri sağlanır
  - Takım çalışması ve iş genişletme olanakları vardır
  - Genel amaçlı, ucuz donanım kullanılabilir

1. Hücresel Üretime Göre Yerleşimin Sakıncaları

- Takım çalışması daha yüksek yetenek gerektirir
- Hücre içi ve hücreler arası yük dengeleme problemleri vardır
- Tampon stoklar gerekebilir
- Ürüne ve sürece göre yerleşimin iyi yönlerini olduğu kadar kötü yönlerini de bünyesinde birleştirmiştir
- Özel amaçlı tezgah kullanma şansını azaltır

2.4. Yeniden Yerleşimde Dikkate Alınması Gereken Hususlar

İşyeri düzenleme, başta ürün ve ürün tasarımındaki değişiklikler olmak üzere, kapasite, malzeme, taşıma, donanım, personel, yapı, servis ve depolama olanakları, yasal düzenlemeler ve teknolojik yenilikler gibi faktörlerin sürekli etkisi altındadır. Bu faktörlerdeki değişiklikler, yerleşim düzenini önemli ölçüde etkilerler. Yeniden düzenlemeye neden olan bu gelişmeleri aşağıdaki gibi sıralamak mümkündür (Su ve Aslan, 1997) :

1. Ürün tasarımındaki değişiklikler
2. Ürün hattına yeni bir ürün eklenmesi ya da çıkartılması
3. Ürüne ilişkin talep artışı ya da düşüşü
4. Pazar konumunun değişmesi
5. Ürün kalitesinin giderek düşmesi
6. Maliyet azaltma çabaları
7. Malzeme taşıma düzeninde aksaklıklar gözlenmesi
8. Ara depolardaki yığılmalar
9. Hasar ve kayıp oranlarındaki artışlar
10. Bakım- onarım maliyetlerinin artması
11. Çalışma koşullarını iyileştirme çabaları
12. Teknolojik yenilikler
13. Donanımın eskimesi ya da kullanım dışı kalması
14. Yüksek kaza oranı
15. Yüksek işçi devri

Bu gelişmeler sonucu ortaya çıkan sorunların çözülmesi için, yapılacak işin niteliğine göre, aşağıdaki davranış biçimlerinden biri benimsenir:

- Mevcut yerleşimde ufak-tefek değişiklikler yapılır
- Yeniden yerleşim yapılır
- Mevcut tesis genişletilerek yeniden düzenleme yapılır
- Yeni bir tesis kurularak yeni düzenlemeler yapılır

Yerleşim düzenlemesinde, üretime doğrudan ve dolaylı olarak katılan her türlü birimin kullanıldığı alanların ve hareket yollarının en iyi biçimde yerleştirilmeleri esas olarak konu edilir. İş yeri düzenlemesi yapılırken aşağıdaki hususlar gözönüne alınır (Su ve Aslan, 1997):

- Malzeme en kısa yoldan taşınmalı, taşıma olabildiğince doğrusal yapılmalıdır
- Malzeme hareketleri tasarlanırken geriye dönüşler en aza indirilmelidir
- Tesis binası, malzeme hareketlerini en aza indirecek şekilde tasarlanmalıdır
- Mevcut binadan en ekonomik şekilde yararlanılmalıdır

- Yapılan planlar esnek olmalı, mümkün gelişmeler önceden tahmin edilmeli, planlar buna göre hazırlanmalıdır

2.5. Tesis Yerleşimi ile İlgili Çalışmalar ve Yeni Eğilimler

Nicel amacı, malzeme taşıma maliyetlerini en küçüklemek; nitel amacı bölümler/tesisler arası yakınlık değerlerini en büyüklemek olan tesis yerleşimi problemleri ile ilgili ilk çalışmalar; Armour ve Buffa, (1963); Hillier, (1963); Hillier ve Conors, (1966); Vollman vd., (1968) ve Drezner, (1980) tarafından sunulan sezgisel yaklaşımlar ve küçük boyutlu problemler için; Gilmore, (1962) ve Lawler (1963) tarafından geliştirilmiş en iyileme yaklaşımları olarak gösterilebilir (Rosenblatt, 1986).

Dinamik tesis yerleşimi problemlerine ilk ciddi yaklaşımın Rosenblatt'ın 1986 tarihli çalışması olduğu söylenebilir (Rosenblatt,1986). Aradan geçen yıllarda konu ile ilgili çalışmalar günümüze kadar artarak devam etmiştir. Meller ve Gau (1996); 1986-1996 yılları arasında yerleşim ile ilgili basılan 91 makaleyi; çalışmada kullanılan model ve algoritmalarına ve Tablo 2.1'de verilen tesis yerleşimi çalışmalarının sınıflandırılması tablosuna göre kategorilere ayırmıştır. Tablo 2.1'de 3 alan mevcuttur. İlk alan, blok yerleşim için geliştirilen tesis yerleşim modelleri ve sezgisellerini içerir. Birkaç çalışma hariç, bu alanı Kusiak ve Heragu (1987) 'nun çalışmasının kapsadığı söylenebilir. İkinci alan, problem tanımına yapılan; zaman boyutu (dinamik yerleşim), belirsizlik (stokastik yerleşim) ya da değerlendirme için birden çok ölçüt (çok kriterli , sağlam ya da esnek) eklentilerini de kapsamaktadır. Üçüncü alan, tesis yerleşim problemlerinin; akış hatları, makine yerleşimi, hücresel yerleşim tasarımı gibi imalat tasarım sistemleriyle bütünleştirilmiş özel durumları içermektedir. Tablo 2.1 aynı zamanda çalışmaların model ya da sezgisel araştırmalarıyla ayrılmasını sağlayan bir düzey de içerir. Ayrıca, A.2.a düzeyi gibi, çalışmanın KAP tabanlı ya da serim (graph-theoritic) tabanlı (bölümler/tesisler arası yakınlık derecelerinin bilindiği varsayılan) olup olmadığı ayrımı da görülebilir.

A kategorisinin sınıflandırılmasında, diğerlerine baskın gelen tek bir yaklaşım mevcut değildir (A.1.a ve A.1.b kategorilerinden 13; A.1.b ve A.2.b kategorilerinden 15 makale mevcuttur). Bu kategorideki serim tabanlı (graph-theoretic) yaklaşımlarla ilgili olarak, asıl çabanın yakınlık ilişkilerinin blok yerleşimine dönüştürülmesine

odaklandığını söylemek mümkündür. Çünkü bu süreç bilgisayarlar için çok zor; insanlar için de çok vakit alıcıdır (Foulds vd., 1986; Meller ve Gau'dan (1996).

Temel modelin uzantılarını içeren B kategorisindeki çalışmaların sayısında gözle görülür biçimde bir artış vardır. İmalat sistemlerinin üzerinde hissedilen değişime uyum sağlama baskısı, tesis yerleşimi ile ilgili yapılan çalışmalarda, Rosenblatt (1986) tarafından ortaya konulan dinamik tesis yerleşim problemleri yönünde eğilime neden olmuştur. Stokastik yerleşim problemleri için, Rosenblatt ve Kropp (1992); amacın beklenen malzeme taşıma maliyetini en küçüklemek olduğu durumlarda; bunun, beklenen akış matrisinden meydana gelen yerleşim problemlerini çözmekle eşdeğer olduğunu göstermişlerdir (Meller ve Gau, 1996). Bu sonuç, bir akış matrisini gözönüne alan geleneksel yerleşim algoritmalarının halen etkili şekilde kullanılabileceğini göstermektedir. Hem stokastik hem de dinamik yerleşim problemlerini içeren bileşik yaklaşımlar birçok araştırmacı tarafından incelenmiştir (EK-1). Bunun nedeni, Rosenblatt ve Lee'nin çalışmalarında belirttiği, beklenen yerleşim maliyetini enküçüklemenin, farklı akış senaryoları için dahi, geçerli olabilecek yerleşimi belirlemeden daha önemsiz olabilmesidir (Meller ve Gau, 1996).

Blok yerleşim problemlerindeki basitlik varsayımlarının aksine özelleştirilmiş yerleşim problemleri olarak adlandırılan C kategorisindeki (özel durumlar) çalışmalar, son 15 yılda artış göstermiştir. Bu problemler, Tompkins ve White ya da Sule tarafından yazılan ders kitaplarında belirtilen ya da KAP'ın farklı varyasyonları olarak modellenen (Pardalos ve Wolkowicz, 1994) "pratik" teknikleri içerirler (Meller ve Gau, 1996). Bu teknikler, bir üretim hattındaki tezgahların yerleşimi çalışmalarını kapsarlar.

Tablo 2.1 ile ilgili olarak eklenebilecek son yorum, "Hücresel Üretim Sistemleri" ile ilgili çalışmaların hızla artmakta olduğudur. Wemmerlov ve Hyer, hücresel imalat teknikleri araştırmaları ve uygulamaları; Mohsen ise yerleşim tasarımında grup teknolojisi çalışmaları ile bu alanda öncü olmuşlardır.

Aradan geçen 10 yılda, konu üzerine yapılan çalışmalar artarak devam etmektedir (Kendall ve Shang, 2006; Dunker vd., 2005; Baykasoğlu ve Gindy, 2001; Balakrishnan ve Cheng, 2000; Lacksonen, 1997).

Tablo 2-1 Tesis Yerleşimi Çalışmalarının Sınıflandırması (Meller ve Gau, 1996)

A.	**Blok Yerleşim için Tesis Yerleşim Modelleri ve Sezgiselleri**
	1. Model
	a. KAP tabanlı
	b. Serim
	c. Karışık tamsayılı programlama
	2. Sezgisel
	a. KAP tabanlı
	b. Serim
	c. Karışık tamsayılı programlama
	d. KAP (sadece eşit alanlı bölümler)
B.	**Tesis Yerleşimi Model Uzantıları**
	1. Dinamik Yerleşim
	a. Model
	b. Sezgisel
	2. Rassal Yerleşim
	a. Model
	b. Sezgisel
	3. Çok ölçütlü, Sağlam ve Esnek Yerleşim
	a. Model
	b. Sezgisel
C.	**Özel Durumlar**
	1. Akış Hatları, Sıra ve Döngü Yerleşimi
	a. Model
	b. Sezgisel
	2. Makine Yerleşimi
	a. Model
	b. Sezgisel
	3. Hücresel Yerleşim
	a. Model
	b. Sezgisel

Konu üzerindeki çalışmaların güncelliğini korumasının sebebi;

- Modüler üretim sistemlerinin uygulamada yaygınlaşmaya başlaması ve dinamik yeniden yerleşimin daha ekonomik bir seçenek haline gelmesi
- Yerleşim problemlerinin de içinde yer aldığı çözümü güç kombinatoryal problemlere başarı ile uygulanabilen üst yordamların (metaheuristics) geliştirilmeye başlanması olarak söylenebilir.

2.6. Bilgisayar Destekli Tesis Planlaması

Bölümler arası akış miktarları, uzaklıklar, yakınlık ilişkileri gibi birçok veri ve matematik işlemi içinde barındıran tesis planlaması sürecinde bilgi teknolojilerinden yararlanma ve bilgisayar kullanımı büyük önem arz etmektedir. Tesis planlamasında aşağıda belirtilen yollar vasıtasıyla bilgisayar desteği sağlanabilmektedir (İşlier, 1997):

1. Bilgisayar destekli tasarım/imalat ortamından yaralanmak
2. Genel amaçlı bir veri tabanındaki verileri kullanmak
3. Matematik programlama, yapay zeka, benzetim gibi başka amaç için geliştirilmiş yazılımlardan faydalanmak
4. Tesis planlaması için özel olarak geliştirilmiş paket program ya da modülleri kullanmak
5. Yeni programlar geliştirmek

2.6.1. Bilgisayar destekli tesis planlama algoritmalarının sınıflandırılması

Bilgisayar destekli tesis planlamasında kullanılan algoritmaların dayandığı dört sınıflandırma kriteri ve kısa açıklamaları aşağıda yer almaktadır (İşlier, 1997):

a. Temel amaca göre sınıflandırma: Yerleşim algoritmaları, kullandıkları öncelik fonksiyonlarına göre sınıflandırılabilmektedir. Bu sınıflar;

- Yeni yerleşimin yapılandırılması (Kurma Esaslı)
- Varolan yerleşim düzenleri üzerinden yeni yerleşim alternatiflerinin geliştirilmesi (Geliştirme Esaslı)

Geliştirme esaslı algoritmalar, analist tarafından sunulan bir başlangıç çözümünü temel alarak amaç fonksiyonunun değerini geliştirmeye çalışırken, kurma esaslı algoritmalar herhangi bir başlangıç çözümüne ihtiyaç duymamaktadır.Bu algoritmalar verilerden hareketle bir yerleşim şekli belirlemektedir.

b. Gereken veri tipine göre sınıflandırma: Tesis planlaması çalışmaları için kullanılan modeller gerek duydukları veri çeşidine göre sınıflandırılabilir.

- **Niteliksel akış verisi**: Faaliyet ilişki çizelgesindekine benzer şekilde sayısalolmayan, ancak bir dönüşüm tablosundan yararlanılarak sayısal değerlere çevrilebilen verilere dayanır.

- **Niceliksel akış verisi**: Bir dönemdeki birim yük taşıma sayısını dikkate alan, sayısal verileri kullanan malzeme akış diyagramı kullanılır.

c. Amaç fonksiyonuna göre sınıflandırma:

- **Uzaklığı esas alan amaç fonksiyonu**: Uzaklığı esas alan amaç fonksiyonu kullanılarak taşıma sayıları ile taşıma mesafelerinin çarpımından oluşan, toplam taşıma maliyetinin en küçüklenmesini hedeflemektedir. Girilen veriler, bölümler arası taşıma sayıları gibi sayısal verilerse bu amaç fonksiyonu kullanılmalıdır.

  Amaç fonksiyonu: $z = \sum_i \sum_j f_{ij} c_{ij} d_{ij}$

  f_{ij} :Birim zamanda i. bölümden j. bölüme taşınan birim yük.

  c_{ij}. i. bölümden j. bölüme birim yük taşıma maliyeti.

  d_{ij}:i. bölümden j. bölüme olan uzaklık olarak gösterilmektedir.

- **Komşuluğu esas alan amaç fonksiyonu**: Faaliyet ilişki çizelgelerine dayalı modeller için, komşuluğu esas alan amaç fonksiyonu kullanılır. Bölümler arası temas uzunluğunu en büyüklemeyi hedeflemektedir.

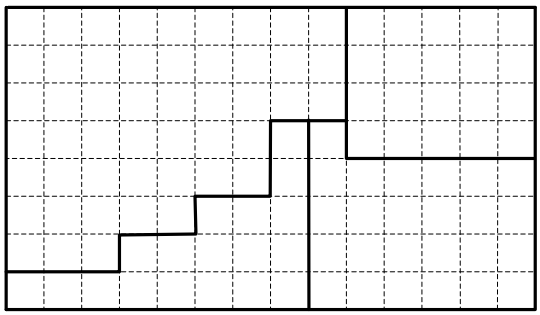
  Amaç fonksiyonu: $enbz = \sum_i \sum_j f_{ij} x_{ij}$

  f_{ij}:Birim zamanda i. bölümden j. bölüme taşınan birim yük.

  x_{ij}: i ve j bölümleri bitişikse 1, değilse 0 değerini alır.

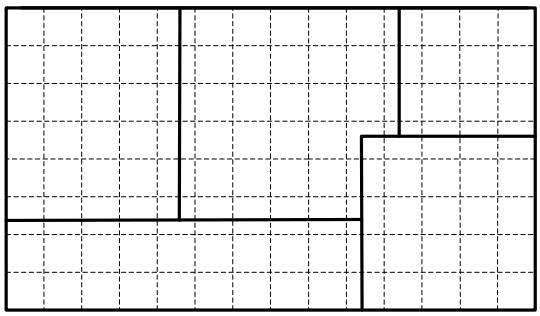
d. Yerleşim gösteriminde kullanılan formatın sınıflandırılması:

- **Kesikli gösterim**: Bu gösterim ile parçalara ayrılmış oturma alanı içerisinde, her bölümün alanı en yakın hücre sayısına yuvarlanmaktadır. Hücrelerin belirlenen boyutuna göre, bölümleri temsil eden hücre sayısı artıp, azalmaktadır (Şekil 2.2).



Şekil 2.2 Kesikli yerleşim gösterimi

- **Sürekli gösterim:** Bu gösterimde, bölümlerin alanları gerçek değerleri ile modele yansıtılır. Bölümleri gösteren çizgiler, hücreleri simgeleyen çizgilerin üstünden geçmek zorunda değildir. Sürekli gösterim, kesikli gösterime oranla daha esnektir, ama bilgisayar ortamına yansıtmak daha zordur. Sürekli gösterimi kullanan algoritmalar dikdörtgen bina ve dikdörtgen bölümleri kullanmakla sınırlandırılmıştır (Şekil 2.3).



Şekil 2.3 Sürekli yerleşim gösterimi

e. Tesisler Arasındaki Uzaklıkları Belirlemek İçin Kullanılan Ölçüm Şekilleri

- **Kuş uçuşu, düz veya Euklid uzaklığı**: Bu ölçüm şekli,iki noktayı birleştiren en kısa yolun bir doğru parçası olduğu kabulüne dayanmaktadır. İki nokta arasında engel olmadığı durumlarda kullanılabilir.
- **Dikdoğrusal (rectilinear) uzaklık**: Şehir içi veya atölye içi yollar gibi birbirini dik olarak kesen yol şebekelerinin kullanılmasını gerektiren durumlarda elverişlidir.

2.6.2. Yerleşim algoritmalarında kullanılan modelleme teknik ve yöntemleri

Tesis yerleşim algoritmalarında kullanılan Craft, Blocplan, Logic ve Multiple modelleme teknik ve yöntemlerinin özellikleri aşağıda özet şeklinde verilmiştir:

- **Craft**

Literatürde bilinen en eski algoritmalardan biridir. 1963 yılında Armour,Buffa ve Vollman tarafından tanıtllmıştır. Diğer algoritmaların gelişmesi için iyi bir platform hazırlamıştır.

Akışlarda veri girişi için malzeme akış diyagramı kullanılır. Yerleşim maliyeti hesaplanırken uzaklığı esas alan amaç fonksiyonu kullanılmaktadır. Craft, tesis içinde bölümlerin belirtilmesinde kesikli gösterimi kullanır. Uzaklık merkezli amaç fonksiyonu değerinin en küçüklenmesi istenmektedir. Bu yöntem geliştirme esaslı olduğu için bir ön yerleşimden yola çıkmaktadır. İyileştirme projeleri için bu ön yerleşim, mevcut durumdur. Yeni bir proje içinse, başka bir yöntemle elde edilmiş bir yerleşim planı bir ön yerleşim olarak kullanılabilir.

- **Blocplan**

Donaghey ve Pire tarafından geliştirilmiştir. Yeni yerleşimin yapılandırılma ve varolan yerleşim düzenleri üzerinden yeni yerleşim alternatiflerinin geliştirilmesi için kullanılabilir. İlişki diyagramı kadar malzeme akış diyagramı, da kullanılır. Ancak iki tablodaki bilgileri birleştirmez, ikisinden birini kullanır. Blocplan sürekli gösterim

kullanır. Maliyetleri hesaplarken; uzaklığı esas alan amaç fonksiyonu ya da komşuluğu esas alan amaç fonksiyonu kullanabilir. Bölümleri bantlar şeklinde düzenler. Bantların genişliğini planlamacı değiştirebilir. Varolan yerleşim düzenleri üzerinde değişiklik yapabilmesi için bütün bölümlerin dikdörtgensel şekilde olması gerekir.

- **Logic**

Logic, Tam tarafından tasarlanmış, tesis yerleşiminin en iyilenmesinde giyotinle kesme yöntemini kullanan bir algoritmadır. Akışlarda veri girişi olarak malzeme akış diyagramını kullanır. Akışlarda yeri girişi olmak malzeme akış diyagramını kullanır. Uzaklığı esas alan amaç fonksıyonu kullanarak yerleşim maliyetini hesaplar.

Yeni yerleşimin yapılandırılması yeya varolan yerleşim düzenleri üzerinden yeni yerleşim seçeneklerinin geliştirilmesi için kullanılabilir. Logic binayı yatay veya dikey parçalara bölme tabanlı çalışır. Her kesimde bölümler kesimlerin kuzey-güney veya doğu¬batı kesimine atanmaktadır. Kesimler ve bölümlerin atanması rassal olarak yapılmaktadır. Dikdörtgensel yapılar içinde dikdörtgensel bölümler oluşturur

Logic alanları eşit olmayan iki bölümü komşuluk şartı aramadan yer değiştirebilir. Blocplan ile benzer algoritmayı kıllanmaktadırlar. Fakat blocplanın çözüm uzayı logic algoritmasının çözüm uzayının alt kümesidir.

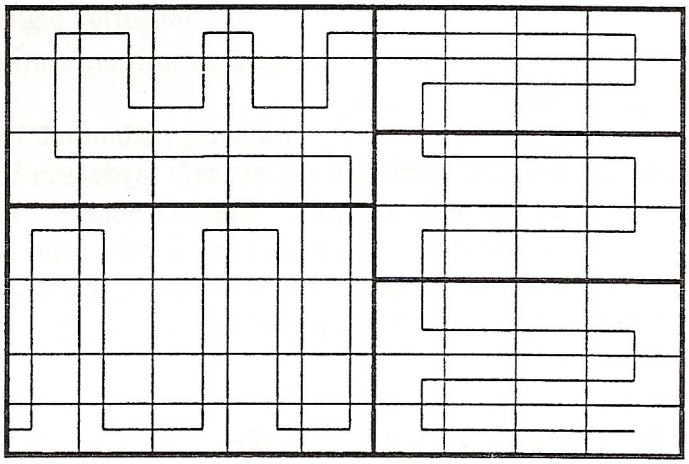
- **Multiple**

Bozer, Meller ve Erlebacher tarafından tasarlanmıştır. Veri girişi olarak malzeme akış diyagramını kullanır. Uzaklığı. esas alan amaç fonksiyonunu kullanarak yerleşim maliyetini hesaplar. Bölümler arası ikili üçlü değiştirmeler yapan en hızlı iniş ilkesini uygulayarak, varolan yerleşim maliyetinde en çok azalmayı sağlayan seçeneği seçer. Craft'tan farklı olarak bölümleri ikili yer değiştirirken komşuluk veya alansal eşitliği aramaz.

Multiple algoritması çalışırken boşluk dolduran eğriler kullanır. Eğri herhangi bir hücreden başlayabilir. Yalnız bir hücreden komşu hücreye yatay veya dikey adımla geçmeli,her hücreyi sadece bir kez ziyaret etmeli ve tüm hücrelerden geçmelidir.

Bölümler hücrelere, bu eğrileri takip eden sırada atanırlar. Alanı eşit olmayan iki bölümün yerlerini değiştirirken, iki bölüm arasındaki bütün bölümler taşınır. Diğer bölümler var olan konumlarında kalır. Eğri bir bölüme atanan bütün hücreleri ziyaret etmeden başka bir bölüme geçemez.

Yerleşim seçenekleri değişik eğriler kullanılarak elde edilebilir. Şekil 2.4'te multiple' de kullanılabilecek eğrilerden biri örnek olarak verilmiştir.



Şekil 2.4 Boşluk dolduran eğriler

Tesis yerleşimi problemlerinin bilgisayar ortamında çözülebilmesi için ilk olarak geliştirilen paket program CRAFT (Bulfa vd., 1964) dır. Ayrıca COFAD (Tompkins ve Reed, 1973,1976), ALDEP (Seehof ve Evans, 1967), CORELAP(Lee ve Moore, 1967; Moore, 1971) yaklaşımları da yerleşim problemlerini bilgisayar ortamında çözebilmek için geliştirilmiş ilk programlardandır. CRAFT ve COFAD toplam malzeme taşıma maliyetini en küçüklemeyi amaçlarken ALDEP ve CORELAP bölümlerin yakınlık oranlarını en büyüklemeyi amaçlar (Rosenblatt, 1986).

Son yıllarda, tesis planlama problemlerini çözmek için geliştirilen paket programların büyük bir kısmı, CAD tabanlı belgelendirme ya da çizim araçlarıdır. Özellikle yerleşim yapısı ya da bu bölümde değinilen geliştirme algoritmaları ile ilgili çözüm içermemektedirler (Tompkins vd., 2003).

Endüstride, diğerlerine göre daha fazla göze çarpan FactoryCAD, FactoryFLOW, FactoryPLAN/OPT ve FactoryVIEW çoklu modüllerini içeren ve Unigraphics Solutions Inc. (UGS) tarafından satışa sunulan VisFactory programdır.

İşyeri yerleşimi, tasarımı ve malzeme akış benzetimi için geliştirilmiş bir ticari program olan eM-Workplace; Tecnomatix Technologies Ltd. tarafından satışa sunulmuştur. PLANOPT ; dikdörtgen şeklindeki bölümlerle çalışan kurma esaslı bir yerleşim algoritması ile çalışır. Bir diğer program, PLANOPT ile aynı firma tarafından satışa sunulan PLANET programıdır (Tompkins vd., 2003).

Tesis yerleşim problemlerini modellemek için farklı teknikler kullanılsa da, olası bölüm çiftleri arasındaki uzaklığı değerlendirmek için yerleşim bölgelerinin merkezlerini kullanan Kareli Atama Problemi (KAP) formülasyonu ile gösterimi Chiang ve Chiang (1998)'e göre de hala en popüler yöntem olarak karşımıza çıkmaktadır**.** İzleyen bölümde, KAP ile ilgili ayrıntılara değinilmiştir.

3. KARELİ ATAMA PROBLEMLERİ

Literatürde Kareli Atama Problemleri (KAP), uzaklık-akış çarpımlarının mümkün olan tüm toplamlarını maliyet olarak alan, yerleşim bölgelerine tesislerin atanmasında, tanımlanan maliyeti en küçüklemeye çalışan bir problem türü olarak tanımlanmaktadır. KAP; tesis yerleşimi, çizelgeleme, gezgin satıcı, elektronikte bağlantı problemleri gibi gerçek hayattaki birçok problemi içine alan, NP-zor problem kategorisinde yer almaktadır. İlk olarak, ekonomik aktivitelerle ilgili matematiksel bir model şeklinde, Koopmans ve Beckmann tarafından ileri sürülmüştür (Loila vd., 2005).

Bu bölümde, literatürdeki KAP ile modellenmiş tesis yerleşimi problemlerinden göze çarpan çalışmalar kısaca özetlendikten sonra KAP'ın en sık kullanılan modelleri, KAP'ın karmaşıklığı ve son olarak, KAP'ı çözmek için geliştirilmiş en iyi çözümü bulan algoritmlar, sezgisel algoritmlar ve üst-sezgiseller (metaheuristics) hakkında bilgi verilmiştir.

3.1. KAP Tabanlı Tesis Yerleşimi Problemleriyle İlgili Çalışmalar

KAP, tesis yerleşim problemlerinde en iyi çözümü elde etmek için geliştirilmiş geleneksel yaklaşımlardan biridir (Meller ve Gau, 1996). Tesis yerleşimi probleminin bu formülasyonunun, orta büyüklüklerdeki problem için (Skorin-Kapov, 1990), kolay kontrol edilemez olması nedeniyle, genetik algoritmalar, tavlama benzetimi ve melez yaklaşımlar gibi çeşitli sezgiseller olumlu ve olumsuz yanlarıyla birlikte Liggett (2000)'in çalışmasında ortaya konmuştur.

Golany ve Rosenblatt (1989), tesis yerleşimi problemlerinin KAP olarak modellenebilmesi için kurma ve geliştirme adımlarından oluşan bir sezgisel geliştirmiştir. CRAFT gibi diğer sezgisel algoritmalarda, algoritmanın ilerlemesi için başlangıç çözümüne ihtiyaç duyulurken bu algoritmada başlangıç yerleşiminin de türetiliyor olması çalışmayı önemli kılan unsurlardandır. Çalışmanın sonucunda, geliştirilen sezgiselin birçok farklı probleme uygulanmasıyla; başlangıç yerleşiminin en iyi çözümü bulmaya çalışmada önem arz ettiği ve simetrik olmayan yerleşim düzenlerinde algoritmanın daha etkili çalıştığı ortaya çıkmıştır.

Lacksonen ve Enscore (1993), dinamik tesis yerleşim problemlerini, farklı bölüm boyutları kullanılabilecek şekilde genişletilmiş KAP ile modellemişlerdir. Çalışmada; CRAFT, kesme düzlemleri, dal sınır algoritması, dinamik programlama ve kesme ağaçları algoritmaları bu model kullanılarak dinamik tesis yerleşimi problemleri için karşılaştırılmıştır.

Kochhar vd. (1998), genetik algoritmalar kullanarak geliştirdikleri HOPE (Heuristically Operated Placement Evolution) tekniğini tesis yerleşim problemlerine uygulamış ve başlangıç çözümüne gerek kalmadan bu teknikle "iyi" çözümler elde ettiklerini belirtmişlerdir.

Chiang ve Chiang (1998), KAP tabanlı tesis yerleşim problemlerini çözmek için zeki yerel arama stratejileri ile ilgili bir çalışma yapmışlardır. Çalışmada; geliştirilen tabu arama, olasılıklı tabu arama, tavlama benzetimi ve melez tabu arama sezgiselleri, bilinen klasik problemlere uygulanarak sonuçlar karşılaştırlmıştır. Çalışma sonucunda elde edilen veriler, tesis tasarımı ve yerleşim problemleri için sezgiselleri kullanmanın rekabetçi avantaj sağaladığını ortaya çıkartmıştır.

3.2. Kareli Atama Problemlerinin Modellenmesi

KAP'ın literatürde kullanılan çeşitli modelleri; tamsayılı doğrusal programlama, karışık tamsayılı doğrusal programlama, grafik, iz ve permütasyonlu gösterim şeklinde sınıflandırılmaktadır (Loila vd., 2005). Bu alt bölümde, en sık kullanılan tamsayılı doğrusal ve karışık tamsayılı doğrusal modelleri ayrıntılı şekilde açıklanmıştır.

3.2.1. Tamsayılı doğrusal programlama

İlk model, ikili kısıtların gevşetildiği doğrusal programlama problemini izleyen 0-1 tamsayılı programlama olarak tanımlanmıştır. KAP'ın en sık kullanılan şekli olan 0-1 tamsayılı formülasyonu Koopmans ve Beckman tarafından ortaya konmuştur. f_{ij}, i ve j bölümleri arasındaki akış ve d_{kp} yerleşim bölgesi k ile yerleşim bölgesi l arasındaki uzaklık olarak belirlendiğinde, amaç fonksiyonu ve kısıtlar aşağıda verildiği gibi modellenir (Loila vd., 2005):

$$\min \quad \sum_{i,j=1}^{n}\sum_{k,p=1}^{n} f_{ij} d_{kp} x_{ik} x_{jp} \tag{3.1}$$

$$k.a \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad 1 \le j \le n, \tag{3.2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad 1 \le i \le n, \tag{3.3}$$

$$x_{ij} \in \{0,1\} \qquad 1 \le i, j \le n. \tag{3.4}$$

Bölümlerin yerleşim noktalarına atanmalarının maliyeti gözönüne alındığında, n sıralı KAP örneğinin genel şekli, $F = \lfloor f_{ij} \rfloor, D = \lfloor d_{kp} \rfloor, B = [b_{ik}]$ olmak üzere üç matrisi içerir. İlk iki matris bölümler arasındaki akış ve yerleşim noktaları arasındaki uzaklığı tanımlarken, B matrisi, bölümlerin yerleşim noktalarına tahsis edilmesinin maliyetini simgeler. Bu problem aşağıdaki gibi tanımlanır:

$$\min \quad \sum_{i,j=1}^{n}\sum_{k,p=1}^{n} f_{ij} d_{kp} x_{ik} x_{jp} + \sum_{i,k=1}^{n} b_{ik} x_{ik} \tag{3.5}$$

$$k.a \quad (3.2), (3.3) ve (3.4)$$

KAP'ın daha genel gösterimi, Lawler tarafından tanımlanmıştır ve akış ve uzaklıkların çarpımıyla uyuşmasına gerek kalmayan c_{ijkp} maliyetini içerir. Lawler formülasyonu aşağıda verilmiştir:

$$\min \quad \sum_{i,j=1}^{n}\sum_{k,p=1}^{n} c_{ijkp} x_{ik} x_{jp} \tag{3.6}$$

$$k.a \quad (3.2), (3.3) ve (3.4)$$

Bu model aynı zamanda, başta Bazaraa ve Elshafei (1979) olmak üzere birçok araştırmacı tarafından kullanılmaktadır (Loila vd., 2005).

3.2.2. Karışık tamsayılı doğrusal programlama

Karışık tamsayılı doğrusal programlama modeli olarak KAP, literatürde farklı şekillerde kullanılmaktadır. Herbirinde kareli ifadeler, doğrusal ifadelere dönüştürülmüştür (Loila vd., 2005). Örneğin, Lawler (1963), aşağıda verildiği gibi n^4 değişken kullanmıştır:

$$c_{ijkp} = f_{ij} d_{kp} \qquad y_{ijkp} = x_{ik} x_{jp}, \qquad 1 \le i, j, k, p \le n.$$

Diğer gösterimlerde, orjinal problemin varyasyonları kullanılmıştır. Genelde, KTDP modelleri tabanlı KAP doğrusallaştırmaları çok fazla sayıda değişken ve kısıt içerdikleri için tercih edilmemektedirler. Ancak bu doğrusallaştırmalar, bazı kısıt gevşetmeleri ile birlikte, en iyi çözüm için daha iyi alt sınırlar elde edilmesini sağlamaktadır. Frieze ve Yadegar (1983) tarafından geliştirilen formül; n^4 gerçek değişken, n^2 0-1 tamsayılı değişken ve $n^4+4n^3+n^2+2n$ adet kısıt içermektedir. Yazarlar, (3.7)-(3.16) arasındaki formüllerin, (3.1)-(3.4) te verilen formüllere eş değer olduğunu göstermiştir (Çela, 1998: Loila vd.'den (2005).

$$\min \quad \sum_{i,j=1}^{n} \sum_{k,p=1}^{n} f_{ij} d_{kp} . y_{ijkp}$$

$$k.a \quad \sum_{i=1}^{n} x_{ik} = 1 \qquad 1 \le k \le n,$$

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad 1 \le i \le n,$$

$$\sum_{i=1}^{n} y_{ijkp} = x_{jp} \qquad 1 \le j, k, p \le n,$$

$$\sum_{j=1}^{n} y_{ijkp} = x_{ik} \qquad 1 \le i, k, p \le n,$$

$$\sum_{i=1}^{n} y_{ijkp} = x_{jp} \qquad 1 \le i, j, p \le n,$$

$$\sum_{p=1}^{n} y_{ijkp} = x_{ik} \qquad 1 \le i, j, k \le n,$$

$$y_{iikk} = x_{iik} \qquad 1 \le i, k \le n,$$

$$x_{ik} \in \{0,1\} \qquad 1 \le i, k \le n,$$

$$0 \le y_{ijkp} \le 1 \qquad 1 \le i, j, k, p \le n.$$

3.3. Kareli Atama Problemlerinin Karmaşıklığı

Literatürdeki çalışmaları birleştiren ortak yargı, KAP'ın en zor kombinatoryal en iyileme problemlerinden biri olduğudur. Sahni ve Gonzales (1976); KAP'ın NP-zor olduğunu göstermiştir. Genellikle, *n>30* boyutundaki KAP problemleri makul bir sürede çözülememektedir. (Loila vd., 2005). KAP'a en iyi çözüm bulmanın yanısıra, en iyi çözüme yaklaşmak da zor olarak nitelendirilmektedir. Sahni ve Gonzales (1976)'in geliştirdiği izleyen iki teorem, KAP'a yaklaşık veya en iyi çözümü bulmanın karmaşıklığını gösterecektir (Burkard vd., 1998; Üstün, 2001):

Teorem 1: KAP, güçlü (strongly) NP- zordur (Burkard vd., 1998).

Tanım: π_0, KAP'ın (A,B) için en iyi çözümü; π_y ise γ algoritması tarafından hesaplanmış çözüm olsun. Bu durumda verilmiş bir $\varepsilon > 0$ reel sayısı için KAP'ın bir γ algoritması, ancak ve ancak herbir *KAP (A,B)* örneği için izleyen eşitsizlik sağlanırsa yaklaşım algoritması olarak isimlendirilir.

$$\left| \frac{Z(A,B,\pi_y) - Z(A,B,\pi_0)}{Z(A,B,\pi_0)} \right| \leq \varepsilon$$

KAP (A,B)'ın ε yaklaşım algoritmasıyla bulunmuş çözümü, ε yaklaşık çözüm olarak isimlendirilir.

Teorem 2: Keyfi bir $\varepsilon > 0$ için, KAP'ın polinom zamanlı ε yaklaşım algoritmasının varlığı P=NP anlamına gelir (Sahni ve Gonzales, 1976: Burkard vd.'den (1998)).

Teorem 2 ve P≠NP genel görüşü gözönünde tutulursa $\varepsilon > 0$ için KAP'a polinom zamanlı ε yaklaşım algoritması bulunması da olası değildir (Üstün, 2001).

3.4. KAP için Çözüm Yöntemleri

Kombinatoryal en iyileme problemlerini çözmek için kullanılan yöntemler en iyi ya da iyi çözümü bulmaya yönelik olarak ikiye ayrılır. En iyi çözümü araştıran algoritmalardan en sık kullanılan stratejiler dal-sınır algoritması ve dinamik programlamadır. Diğer yandan, farklı kavramları kullanan çok sayıda sezgisel teknik

bulunmaktadır. İzleyen alt bölümde, Loila vd. (2005) ‘nın çalışmasında ayrıntılı olarak değinilen bu yaklaşımlara yer verilmiştir.

3.4.1. Eniyileme algoritmaları

KAP’ta bütünsel en iyiye ulaşmak için; dal-sınır algoritması, kesme düzlemleri, dinamik programlama ile bu yöntemlerin dal-sınır/kesme düzlemleri gibi kombinasyonlarını içeren farklı yöntemler kullanılmaktadır.

Dal-sınır algoritması bu yöntemler içinde en çok bilinen ve kullanılan algoritmadır. Problemin alt sınırlarını tanımlayan tahsis ve kesme kurallarıyla belirlenir. Son yıllarda, dal-sınır teknikleriyle paralel uygulamayı birleştiren yordamlar geniş bir uygulama alanı bulmuşlardır. Bu birleşim yöntemi sonucu, KAP’ın çözümündeki en iyi sonuçlara ulaşılmıştır. Ancak, büyük problemlerin çözümündeki başarının, donanım teknolojilerinin gelişimine bağlı olduğu da unutulmamalıdır .

Dinamik programlama tekniği, akış matrisinin, bir ağacın yakınlık matrisi olduğu KAP’ın özel durumları için kullanılmaktadır. Christofides ve Benavent (1989), KTDP yaklaşımını gevşetilmiş problemlere uygulayarak bu özel durumlarla ilgili çalışmalar yapmışlardır .

Kesme düzlemleri yöntemi, Bazaraa ve Shearali (1980) tarafından ortaya konmuş ve bu yöntemle başlangıçta çok başarılı sonuçlar elde edememişlerdir. Bu nedenle, KTDP ve Benders adı verilen ayrıştırmayı içeren bazı sezgiselleri bu yöntemin içine katmışlardır. Bu yöntem, şimdiye kadar çok fazla alanda uygulanmamıştır. Ancak bu yöntemle, KAP için iyi kaliteli çözümler elde edilmiştir. Bu yöntemin yavaş yakınsama özelliği, yöntemi sadece küçük boyutlu problemlere uygulama imkanı vermektedir .

3.4.2. Sezgisel algoritmalar

Sezgisel algoritmalar, en iyi .çözümü elde etmeyi garanti etmezler. KAP için kullanılan sezgisel yöntemlerden en sık kullanılan kurma yöntemi (constructive method), sınırlı sayımlama (limited enumaration) ve iyileştirme yöntemiyle (improvement methods) ilgili ayrıntılar aşağıda verilmiştir.

Algoritmanın herbir iterasyonunda, uygun çözüm gibi bir permütasyon kullanan kurma yöntemi Gilmore (1962) tarafından geliştirilmiştir. Bu yöntemde, A ve L kümeleri tanımlanmıştır. A kümesi, yerleştirilen bölüm/tesisleri, L kümesi ise ilgili yerleşim bölgelerini tanımlamaktadır ve başlangıçta her ikisi de boş kümedir. π permütasyonu sezgisel olarak kurulur ve her bir adımda $i \notin A, j \notin L$ ve $\pi(i) = j$ olacak şekilde (i,j) nin yeni atamaları seçilir. N boyutlu bir örnek için, problemin tüm permütasyonları deneninceye kadar süreç tekrarlanır.

Sayımlama yöntemi, en iyi çözümü ancak, sayımlama sürecinin en son adımına ulaşılabildiğinde garanti eder. Bununla birlikte,sayımlamanın başında, iyi hatta en iyi çözümü bulmak mümkündür. Sayımlamayı yönlendiren bilgi ne kadar iyi olursa, iyi kaliteli çözümlerin erken bulunma şansı da o kadar fazla olur. Bunula birlikte, en iyiyi garantilemek çok daha fazla zaman alır.

İyileştirme yöntemleri, yerel arama algoritmaları ile benzerlik gösterirler. QAP sezgisellerinin çoğu bu kategoridedir. Bir geliştirme yöntemi uygun bir çözümle başlar ve komşuluğundaki diğer çözümleri arayarak bu çözümü geliştirmeye çalışır. Bu süreç, herhangi bir geliştirme kalmayıncaya kadar devam eder. Bu yöntemin temel elementi, komşuluk ve komşulukların hangi sırayla analiz edileceğini belirleyen seçim kriteridir.

3.4.3. Üstsezgiseller (Metaheuristics)

1980'lerin başlangıcında, kombinatoryal eniyileme problemleri için önerilen sezgisellerin çoğu probleme özeldi. Bir süre sonra, daha genel teknikler olan üstsezgiseller ortaya çıkmaya başladı. Bunlar, problem yapısına uyarlanan stratejiler olarak nitelendirilebilirler. Üstsezgisellerin ortaya çıkması, KAP ile ilgili çalışmaların hız kazanmasına neden olmuştur.

Tavlama benzetimi, istatistiksel termodinamikle kombinatoryal eniyileme algoritmaları arasındaki analojiyi kullanan bir yerel arama algoritmasıdır. Bu analoji, kombinatoryal eniyileme problemlerinin uygun çözümleriyle fiziksel sistemin düzeyleri ilişkilendirilerek yapılır. Fiziksel sistemin enerji düzeyleri, maliyetlerle ilişkilendirilir. E_i *ve* E_{i+1} iki komşu çözümü temsil eden 2 enerji düzeyi ve $\Delta E = E_{i+}1 - Ei$ olsun. Eğer,

- $\Delta E < 0$ ise, enerji azalması olur ve süreç devam eder. Başka bir deyişle, problemin maliyet fonksiyonunda azalma maydana gelir ve yeni atama kabul edilebilir

- $\Delta E = 0$ ise sabit bir durum sözkonusudur ve enerji düzeyinde bir değişim olmaz. Bu, problemin maliyet fonksiyonunun değişmediğini gösterir

- $\Delta E > 0$ durumu, enerji düzeyinde artış olarak nitelendirilir ve problemin maliyet fonksiyonunun artışını ifade eder. Bu durumda yeni atama kabul edilmez.

Burkard ve Rendl (1984), KAP'ın ilk tavlama benzetimi uygulamalarından birini gerçekleştirmişlerdir (Loila vd., 2005).

Genetik algoritmalar (GA), doğadaki doğal seçim ve uyum sağlamayı taklit ederler. GA'lar, genellikle rassal olarak türetilen bireylerden (çözümler) oluşan bir populasyonla başlarlar. Populasyon içindeki bireylerin uyum değerleri, başka bir deyişle maliyetler, değerlendirilerek en düşük maliyetli çözümler seçilir. Bu alt kümeye, yeni bir alt küme (populasyon) oluşturmak için genetik operatörler uygulanır. Bu süreç durma ölçütüne kadar tekrar edilir. KAP çözümlerinde genetik algoritmaların kullanılması için farklı fikirler Tate ve Smith (1995), Drezner (2005) gibi birçok araştırmacının çalışmalarında ortaya konmuştur KAP için GA'nın kullanımında, küçük boyutlu problmelerde bile bazı zorluklarla karşılaşılmıştır. Bu sorun, bazı melez uygulamalarla giderilip etkili sonuçlar elde edilmiştir (Loila vd., 2005).

Dağıtık arama (Scatter search), Glover (1977) tarafından tamsayılı doğrusal programlama problemleri için geliştirilen sezgisel bir çalışmadır. Yeni çözüm vektörleri türetmek için, çözüm vektörlerinin doğrusal bileşimlerini alan evrimsel bir tekniktir. Bu üstsezgisel, başalngıç ve evrimsel adımlardan oluşur. Başlangıç adımında, iyi çözümlerin bir kümesi (referans) oluşturulur. Her izleyen adımda (evrimsel adım), referans alt kümenin seçilen stratejik kombinasyonları kullanılarak yeni çözümler türetilir (Loila vd., 2005).

Karınca kolonisi en iyilemesi, karıncaların yiyecek bulmak için izledikleri yoldan esinlenerek geliştirilmiştir. Karınca kümesi, bir problemi çözmek için olağan bir

işbirliği içindedirler. Bu yöntemin başlıca özelliği, bu ajanların etkileşiminin sinerjik bir etki yarattığı gerçeğidir. Çünkü birbirleriyle etkileşim halinde bulunan bu ajanlar birlikte çalıştıklarında, elde edilen çözümün kalitesi artmaktadır. Dorigo vd. (1996) ve başta olmak üzere, son yıllarda yapılan çalışmalar; yakın iyi çözümlere sahip örnekler başta olmak üzere, karınca kolonisi eni iyliemesinin diğer üstsezgiseller karşısında rekabetçi üstünlüğe sahip olduğunu göstermiştir. Karınca kolonisi en iyilemesinin KAP'a uygulanmasını içeren en önemli çalışmaların başında Gambardella vd. (1999)'ın olduğu söylenebilir.

Tabu arama, Glover (1989), tarafından geliştirilmiş bir yerel arama algoritmasıdır. Özelliği, arama sürecinde yer alan kötü çözümlerin sürekli güncellenen bir listesini tutmasıdır. Her bir çözüm, bir öncelik değeri ya da amaç ölçütüne sahiptir. Temel içeriği, arama sürecinin değerlendirme geçmişinin tutulduğu tabu listesidir. Tabu aramanın KAP uygulamalarına Skorin-Kapov (1990,1994), Taillard (1991) örnek olarak verilebilir. Bu algoritmanın başarısının tutulan tabu listesinin büyüklğüne ve bu listenin yönetilme şekline bağlı olması olumsuz bir durum gibi görünmesine rağmen, KAP için çözüm bulmada oldukça etkili stratejiler içermektedir (Loila vd., 2005) .

GRASP (Greedy Randomized Adaptive Search Procedure), her bir adımında, problemin bir yaklaşık çözümünü elde eden rassal ve iteratif bir tekniktir. Son çözüm, tüm iterasyonlar arasındaki en iyi sonuçtur. Her adımda, ilk çözüm, rassal bir açgözlü fonksiyon ile kurulduktan sonra izleyen çözümler, önceki çözümün yerel bir arama algoritmasına uygulanmasıyla elde edilir. Bu teknik, çeşitli araştırmacılar tarafından KAP'a uygulanmıştır (Loila vd., 2005).

VNS (Variable Neighborhood Search), bir komşuluk kümesi içindeki uygun şekilde tanımlanan sistematik hareketler üzerine kurulmuştur. Birçok değişim kuralına başvurulabilir. Değişim, keşfedilen komşuluk daha iyi çözümü üretmeyene kadar uygulanır. Taillard ve Gambardella (1999)'nın çalışmasında 3 adet VNS stratejisi ortaya konmuştur (Loila vd., 2005

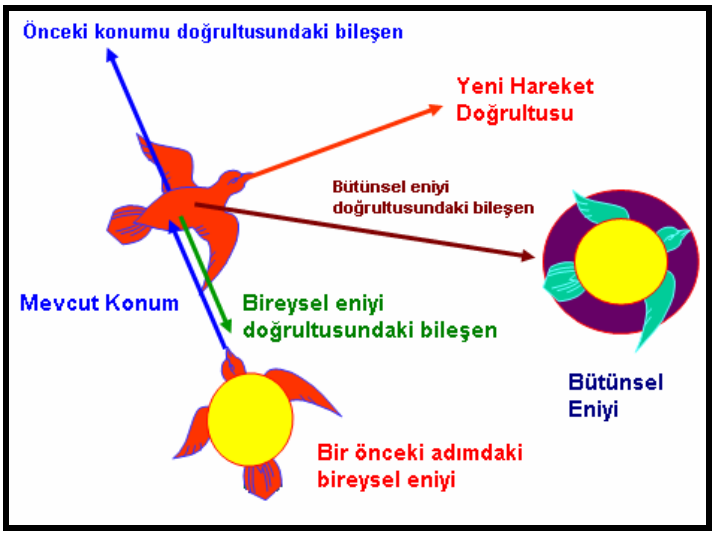
4. TAKIM ZEKASI TABANLI RASSAL ENİYİLEME (TZRE) ALGORİTMASI

Populasyon tabanlı algoritmalar, genetik algoritmalar ile eniyileme alanında yer edinmeye başlamış, daha sonra evrimsel algoritmalar ve karınca-kolonileri ile yaygınlaşmıştır. Bu kuşak algoritmaların bir kısmı –ve literaturde en yeni olanı- sosyal zeka (swarm intelligence) üst başlığında incelenen özel bir yaklaşımdır. Sosyal zeka ve potansiyel uygulamaları 90'lı yılların başlarında Dorigo vd. (1996) 'nun karınca kolonisi eniyilemesi; Kennedy ve Eberhart (1995)'ın takım zekası tabanlı rassal eniyileme algoritmasıyla ilgili ilk yayınları ile tanınmaya başlanmıştır. Takım zekası tabanlı eniyileme, karşılıklı etkileşim halinde bir toplumu oluşturan bireylerin – takım olarak anılan - ortak bir problemin çözümüne kendi yerel çözümleri ile yaklaşmakla birlikte aynı topluluğun diğer bireylerinin elde ettiği iyi çözümlerden de etkilenerek aynı zamanda kendi çözümleri ile de topluluğu etkileyerek bütünsel (global) çözüme katkıda bulunmaları esasına dayanır. Özellikleri ilham kaynağı olan topluluğun (balık, karınca, böcek, kuş, insan vb.) sosyal iletişim ve yaklaşımlarına bağlı olarak şekillendirilmektedir.

İzleyen alt bölümlerde sırasıyla; TZRE'nin ortaya çıkışı ve gelişimi, temel kavramları, kullanılan parametreler ve akış şeması, kontrol parametrelerinin seçimi ve TZRE'nin uygulama alanları ile mevcut çalışmalara yer verilmiştir.

4.1. TZRE'nin Ortaya Çıkışı

Canlılar bazen sürü şeklinde yaşarlar. Yapay yaşamla ilgili araştırma yapan bilim adamları canlıların sürü halindeki davranışlarını inceleyerek sürü modellerini bilgisayar ortamında tekrar biçimlendirmektedirler. Kuş ve balık sürüleri gibi birlikte hareket eden canlıların davranışları birkaç basit kural ile modellenebilir. Ancak, her bireyin (ajanın) davranış kuralları basit olsa da sürünün hareketi karmaşık olabilir. Sürünün içindeki her bireyin hareketlerinin basit kurallarla belirlenmesi bunların basit vektörlerle modellenmesini de olanaklı kılmaktadır (Şekil 4.1). Bu özellik; TZRE'nin temel özelliklerinden biridir



Şekil 4.1 Sürü içindeki bireyin hareket bileşenleri (El-Sharkawi, 2006)

Boyd ve Richerson (1985), insanların düşünme ve karar verme sürecini inceleyerek *bireysel öğrenme* ve *kültürel iletim* kavramlarını ortaya koymuşlardır. Gözlemlerine göre, insanlar karar verme sürecinde önemli iki tür bilgi kullanmaktadırlar. Sözü edilen ilk bilgi, seçeneklerini denedikleri, hangi durumun daha iyi olduğunu bildikleri kendi deneyimleridir. İkinci bilgi, çevrelerindeki diğer bireylerin nasıl yaptıklarını içeren yöntem bilgisi, diğer bir deyişle, diğer insanların deneyimleridir. Yani, onlar komşularının şimdiye kadar en çok olumlu buldukları seçenekleri biliyorlar ve en iyi davranışın nasıl olumlu bir etki yaratacağının farkındalar. Özetle, her bir birey, kararını kendi deneyimleri ve diğer insanların deneyimleri doğrultusunda vermektedir. Bu özellik, TZRE'nin ikinci temel özelliği olarak karşımıza çıkmaktadır. (Fukuyama, 2006).

4.2. TZRE'nin Temel İlkeleri

Kennedy ve Eberhart, kuş sürüleri gibi organizmaların sosyal davranışlarını inceleyerek türlerin bir grup halinde hedeflere ulaşmada yaptıkları işbirliğinin etkilerini araştırmışlar ve TZRE kavramını ortaya koymuşlardır.

TZRE, kuş ve balık sürülerinin sosyal davranışlarını taklit etmeye çalışan işbirlikçi populasyon tabanlı bir arama gerçekleştirmektedir (Kennedy vd., 2001). TZRE, keşfetme ve yararlanmayı dengelemek amacıyla yerel arama yöntemleri (bireysel

deneyim aracılığıyla) ile bütünsel arama yöntemlerini (komşuluk deneyimi aracılığıyla) birleştirir (Salman vd., 2002). Geçtiğimiz birkaç yıl içinde, TZRE, birçok araştırma ve uygulama alanlarına, diğer yöntemlere göre daha yüksek başarıyla uygulanmıştır. (Hu, 2006). TZRE'da, aday çözümler, birey (particle) olarak adlandırılırlar, mevcut en iyi bireyleri izleyerek problem uzayı boyunca arama yaparlar. Bireyler o ana kadar buldukları en iyi çözümlerinin (uyum değeri) koordinatlarını hatırlarlar, aynı zamanda, komşuluklarının ya da bazen tüm populasyonun o ana kadar buldukları en iyi çözüme doğru çekilirler. TZRE bireylere aynı zamanda; bireylerin o zamana kadar bulunan yerel ve bütünsel en iyi çözümlerinin koordinatları doğrultusundaki ivmelerini belirleyen rassal ağırlıklandırılmış hızları atar. Bu nedenle, arama uzayındaki bireylerin durumları; ***konumları*** ve ***hızlarıyla*** belirlenir. Herbir bireyin pozisyonu XY ekseninde tariflenebilir. Aynı zamanda hızları da v_x (X ekseni doğrultusundaki hızı) ve v_y (Y ekseni doğrultusundaki hızı) olarak belirlenebilir.

Sürü halinde hareket eden bireyler amaç fonksiyonunu eniyilemeye çalışırlar. Her birey o ana kadarki eniyi değerini *(pbest*) ve konumunu bilir. Bu bilgi, herbir bireyin bireysel deneyimini oluşturur. Bunun da ötesinde, herbir birey, *pbest*ler arasındaki şimdiye kadar elde edilen en iyi değeri *(gbest)* de bilir. Bu bilgi, çevresindeki diğer bireylerin başarılarından bireyin haberdar olmasını sağlayarak bütünsel deneyimi oluşturur.

Herbiri, aşağıdaki bilgileri izleyerek konumlarını geliştirir:

- Mevcut konumları (x, y)
- Mevcut hızları (v_x, v_y)
- *pbest* ve mevcut konumları arasındaki uzaklık
- *gbest* ve mevcut konumları arasındaki uzaklık

Bu gelişim, *hız* kavramıyla tariflenebilir. Kennedy ve Eberhart, bireylerin hız ve konumları için izleyen denklemleri geliştirmiştir (Kennedy vd., 2001):

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)) \quad (4.1)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (4.2)$$

Bu denklemlerde;

$\vec{x}_i$, i. bireyin konumu,

$\vec{v}_i$, i. bireyin hızı,

$\vec{p}_i$, i.bireyin o ana kadar ulaştığı bireysel en iyi konumu,

$\vec{p}_g$ o ana kadar bireylerin ulaştığı bütünsel en iyi konumu,

φ , rassal sayı,

t , zamanı temsil etmektedir.

Eşitlik (4.1), bireyin önceki hızı ile bireysel ve bütünsel en iyi konumu arasındaki uzaklıklarına göre yeni hızı hesaplamak için kullanılır. Daha sonra birey, Eşitlik (4.2)'ye göre yeni pozisyonuna doğru uçar. Herbir bireyin başarısı, problemle iligli maliyet fonksiyonu ile orantılı olan, daha önceden tanımlanmış bir uyum değeri ile ölçülür. Bu süreç, kullanıcının tanımladığı durma ölçütüne kadar devam eder.

Birey, Eşitlik (4.1) ve Eşitlik (4.2) nin etkisiyle, iki "eniyi" nin ağırlıklı ortalaması şeklinde tariflenen ve aşağıda verilen nokta civarında eğriler çizerek hareket eder:

$$\frac{\varphi_i \vec{p}_i + \varphi_2 \vec{p}_g}{\varphi_1 + \varphi_2} \quad (4.3)$$

Bu sistem, birtakım yöntemlerle hız yavaşlatılmadığı sürece, giderek genişleyen salınımlarla arama alanını anlamsızlaştırcak düzeyde genişleme eğilimindedir. Kennedy vd., (2001), bu eğilimi önlemek için en büyük hız (v_{max}) parametresini tariflemişlerdir. d boyutlu uzayda, i. birey için aşağıdaki denklemler geliştirilmiştir:

Eğer, $v_i > V_{max}$ ise $v_i = V_{max}$

eğer, $v_i < -V_{max}$ ise $v_i = -V_{max}$

V_{max}'ın tariflenmesi, sürü bir noktaya yakınsama ya da bozulma eğilimi göstermese de, bireylerin sınırlar arasında salınımına izin verir. Sürü, yakınsama olmadan, salınımlarıyla, optimum alanda iyi noktalara (çözümlere) ulaşır.

Sürünün arama hareketini kontrol eden yöntemlerden biri *"atalet ağırlığı" (inertia weight)* uygulamasıdır. Diğer bir yöntem, Fransız matematikçi Maurice Clerc'in geliştirdiği, formüle farklı ifadelerle yansıtılabilen *"hızlandırma katsayıları" (acceleration coefficients)* nı içerir. Atalet ağırlığı ve hızlandırma katsayılarının Eşitlik (4.1)'e yansıtılmış şekli, diğer bir deyişle formülün ayrıntılandırılmış şekli, aşağıda verilmiştir (Kennedy vd., 2001):

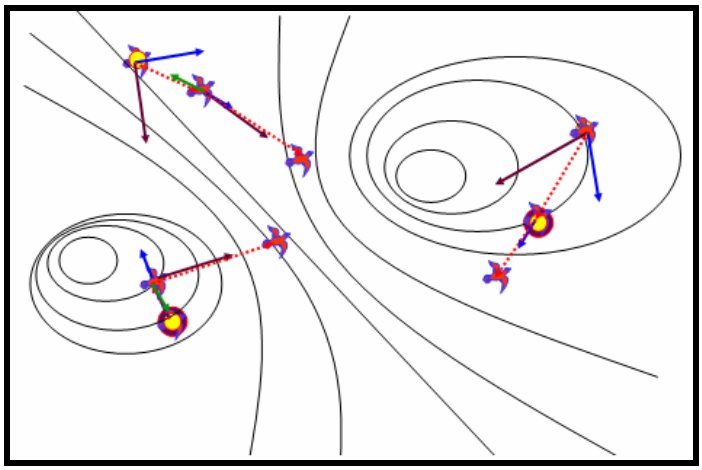
$$\vec{v}_i(t) = w\vec{v}_i(t-1) + c_1 rnd()(\vec{p}_i - \vec{x}_i(t-1)) + c_2 Rnd()(\vec{p}_g - \vec{x}_i(t-1)) \quad (4.4)$$

Eşitlik (4.4)'te, Eşitlik (4.1)'den farklı olarak, w atalet ağırlığını; , rnd ve Rnd rassal sayıları; c_1 ve c_2 hızlandırma katsayılarını göstermektedir.

Küçük atalet ağırlığı, mevcut arama alanında ince ayarlamaya doğru baskı uygularken daha büyük atalet ağırlığı, bireyleri bütünsel keşfe (yeni alanların aranması) doğru yönlendirir. Atalet ağırlığı ve hızlandırma katsayılarının uygun seçilmesi, bütünsel ve yerel arama arasında denge kurmayı sağlar.

Eşitlik (4.4)'ün sağ tarafı 3 terimden oluşur. İlk terim, partiklün önceki hızıdır. İki ve üçüncü terimler, partiklün hızını değiştirmek için kullanılırlar. Son iki terim olmadan birey, arama alanının sınırlarına çarpıncaya kadar, aynı doğrultuda uçmaya devam eder. Yani, yeni alanlar keşfetmeye çalışır. Bu nedenle ilk terim arama sürecinin yönünü değiştirir (diversification). Öte yandan, ilk terim olmadan, uçan bireyin hızı sadece mevcut konumu ve geçmişteki en iyi konumuna göre belirlenir. Diğer bir deyişle, birey kendi pbest ve gbestine yakınsamaya çalışacaktır. Bu nedenle, iki ve üçüncü terimler arama süreci içeirsinde, yoğunlaşma (intensification) ile bağdaştırılmışlardır (Fukuyama, 2006).

Şekil 4.2, sürü içindeki bireylerin çözüm alanındaki arama sürecini; yerel ve bütünsel en iyilerden etkileniş doğrultularını göstermektedir.



Şekil 4.2 Çözüm alanında bireylerin arama hareketleri (El-Sharkawi, 2006)

TZRE'nin genel akış şeması aşağıda verilmiştir:

1. **Adım:** Herbir bireyin başlangıç durumlarının belirlenmesi

   - t = 0 anında, bireylerin ilk konum ve hızları, izin verilen aralıkta rassal olarak türetilir
   - Her bir bireyin ilk konumları, *pbest* olarak atanır
   - *pbest* lerin en iyi değeri *gbest* olarak atanır ve en iyi değere sahip birey numarası tutulur

2. **Adım:** Herbir bireyin konumunun değerlendirilmesi

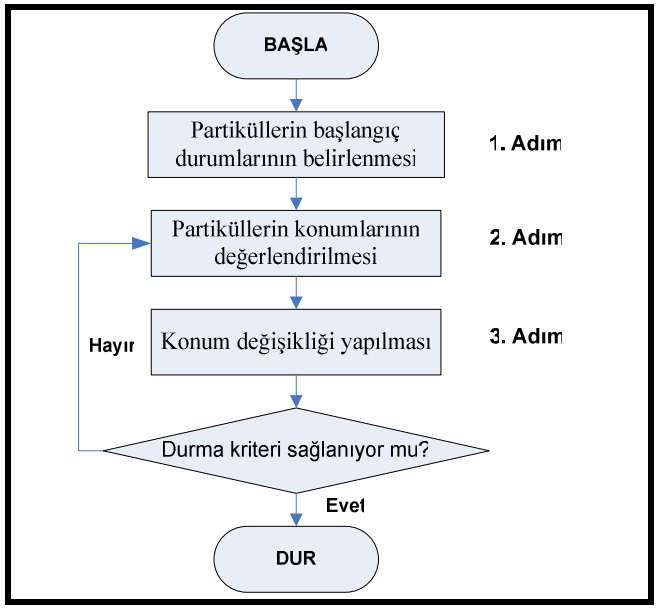
   - Tüm bireyler için amaç fonksiyonu hesaplanır
   - Hesaplanan amaç fonksiyonu mevcut *pbest*ten daha iyi bir değere sahipse *pbest*, mevcut değerle değiştirilir
   - *pbest* in en iyi değeri *gbest*ten daha iyiyse, *gbest* en iyi değerle güncellenir ve en iyi değere sahip paritkül numarası tutulur

3. Adım: Konum değişikliği

- Herbir bireyin mevcut konumları, Eşitlik (4.1) ve Eşitlik (4.2) kullanılarak değiştirilir.

4. Adım: Durma ölçütünün kontrolü

- Gerçekleşen tekrar sayısı, önceden belirlenmiş iterasyon numarasına eşitse çık; değilse ikinci adıma git



Şekil 4.3 TZRE'nin genel akış şeması

Şekil 4.3'te, TZRE'nin genel akış şeması gösterilmiştir. TZRE'nin arama sürecinin özellikleri aşağıdaki gibi özetlenebilir:

1. TZRE'de, genetik algoritmalarda olduğu gibi birden çok arama noktası kullanılır ve arama noktaları, pbestler ve gbestler kullanılarak en iyi noktaya yavaş yavaş yaklaşır.

2. Eşitlik (4.1)'in sağ tarafının ilk terimi, yön değişimine karşı gelir. İkinci ve üçüncü terimleri, arama sürecinin yoğunlaşmasına karşılık gelir. Aslında bu yöntem, arama sürecinde yön değişimi (bütünsel arama) ve yoğunlaşmayı

(yerel arama) etkili şekilde kullanan bir dengeleme (well-balanced) mekanizmasıdır.

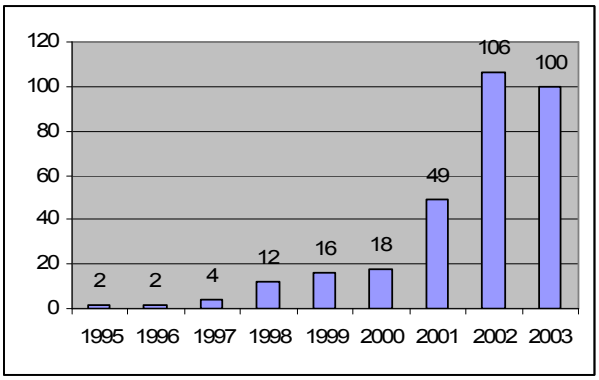
3. Akış şeması, iki boyutlu uzay için açıklanmıştır. Bununla birlikte, bu süreç, kolayca n-boyutlu uzaya da uygulanabilir.

4.3. Parametre Seçimi ve Yakınsama Analizi

TZRE'nin başarısını arttırmak amacıyla, populasyon tabanlı bu algoritmanın yakınsama özelliklerinin daha iyi anlaşılabilmesini konu eden çok sayıda çalışma yapılmıştır (Trelea, 2003). TZRE kontrol parametreleri; hızlandırma katsayıları *(c_1 ve c_2)*, atalet ağırlığı *(w)*, hız kısıtı *(v_{max})* ve sürü büyüklüğü *(P)* dür (Bergh and Engelbrecht, 2005). TZRE'nin başarısı, atalet ağırlığı, hızlandırma katsayıları, hız kısıtı ve sürü büyüklüğünün uygun seçimine bağlıdır. Uygun olmayan başlangıç kontrol parametrelerinin seçimi, çözümde sapmalara ve yerel en iyi çözümlere sebep olabilir. Bergh ve Engelbrecht (2005) gibi bazı araştırmacılar bu parametrelerin belirlenmesi üzerine analitik çalışmalar yapmışlardır. Ancak, her bir özel uygulamada, sürü içinde hareket eden bireylerin davranışlarını daha anlaşılır kılabilmek için, kaçınılmaz olarak deneysel çalışmaya gerek duyulmaktadır. Kontrol parametrelerinin seçimi ve yakınsama özellikleri ile ilgili ayrıntılar Bölüm 5'te açıklanmıştır.

4.4. TZRE'nin Uygulama Alanları ve Yapılmış Çalışmalar

Kennedy ve Eberhart'ın 1995 yılındaki çalışmalarıyla ortaya çıkan TZRE, göreceli yeni bir evrimsel paradigma olarak son birkaç yılda hızla gelişmiş ve bu yeni yaklaşımla ilgili, 2003 yılı sonu itibariyle 300'ün üstünde makale yayınlanmıştır. Bu yeni algoritma ile ilgilenen araştırmacıların sayısı hızla artmakta; ve yapılan çalışmalar farklı bakış açıları içermektedirler. Şekil 4.4; 1995-2003 yılları arasında yayımlanmış çalışmaların dağılımını göstermektedir (Hu vd., 2004):



Şekil 4.4 Yıllar bazında yayımlanmış çalışma sayıları

TZRE, ortaya çıkışından bu yana çeşitli alanlarda uygulanmaktadır. Bazı uygulamalar genel, bazıları daha özel çalışmalardır. Kennedy vd. (2001), TZRE'nin ilk uygulamalarına çalışmalarında geniş ölçüde yer vermiştir. Literatürde, başta Endüstri Mühendisliği ile ilgili olmak üzere, en çok göze çarpan uygulama alanları aşağıda başlıklar halinde sıralanmıştır:

- Gezgin satıcı problemleri
- Çok amaçlı esnek atölye çizelgeleme problemleri
- Görev atama problemleri
- Sinir ağları öğrenme algoritmaları
- İnsanların titreme analizi (Humor tremor analysis)
- Bulanık sinir ağlarında kural türetme
- Sayısal denetimli şekil vermede eniyileme
- Reaktif güç ve voltaj kontrolü
- Dağıtım durum tahmini
- Güç sistemleri sabitleyici tasarımı

Clerc (2000), TZRE'nin gücünü ve sınırlarını daha iyi belirleyebilmek amacıyla, bu tür en iyileme sezgisellerinde çok kötü sonuçlar alınan gezgin satıcı problemi ile ilgili bir çalışma yapmıştır.

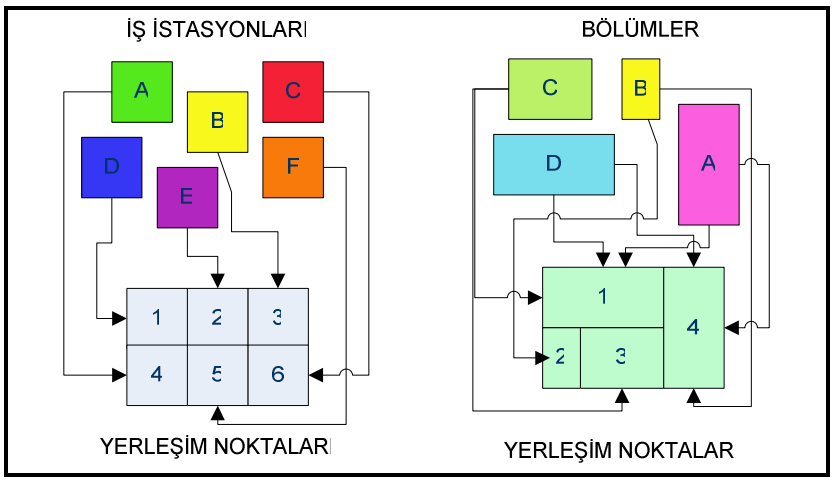
Xia ve Wu (2005), TZRE ve tavlama benzetimi içeren melez bir yaklaşımı, çok amaçlı esnek atölye tipi çizelgleme problemlerine uygulamış ve önerilen yaklaşımın bu tür problemler için uygulanabilir olduğunu ortaya koymuştur.

TZRE ile ilgili yapılmış mevcut çalışmalar incelendiğinde, daha çok elektronik alanındaki uygulamalar göze çarpmaktadır. Uygulama alanı olarak, Endüstri Mühendisliği problemlerine çok fazla değinilmemiş olması, TZRE'nin EM problemlerinde yaygın olarak kullanılacak bir en iyileme aracı olabileceğini göstermektedir.

İzleyen bölümde, KAP yapısında modellenmiş tesis yerleşim problemleri için önerilen Takım Zekası Tabanlı Rassal Eniyileme Algoritması ve yapılan geniş çaplı deneysel çalışma açıklanmıştır.

5. KAP TABANLI TESİS YERLEŞİM PROBLEMLERİ İÇİN TAKIM ZEKASI TABANLI BİR RASSAL ENİYİLEME ALGORİTMASI

Bir tesiste, bölümlerin yerleşim noktalarına atanması, çoğu zaman sürecin tamamının eksiksiz olmasına yetmese de, modüler üretim teknolojilerinin sağladığı olanaklardan etken bir şekilde yararlanmak açısından büyük önem taşımaktadır.



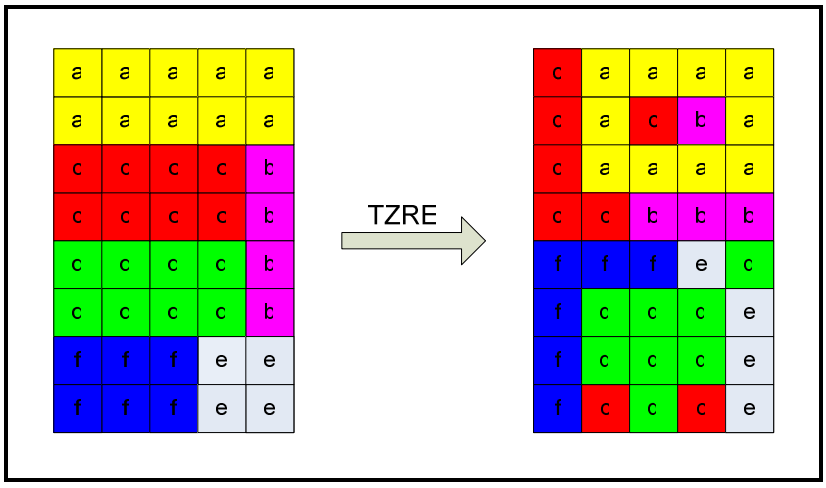
Şekil 5.1 Boyutları eşit olan/olmayan iş istasyonları ve bölümlerin yerleşim noktalarına atanmasının gösterimi

Bölümlerin yerleşim noktalarına atanmasından sonraki adım, bölümler içindeki iş istasyonlarının ilgili yerleşim noktalarına atanmasıdır. Bazen bu adım, ilk atama adımının ikinci geçişi olarak düşünülebilir. Sözü edilen tekrarlı; çok geçişli tesis yerleşim süreci; Bölüm 3.4'te ayrıntılı şekilde anlatılan dal-sınır algoritması, kesme düzlemleri gibi kapalı ya da açık sayımlama tabanlı algoritmalardan farklı, etkin bir çözüm algoritmasına ihtiyaç duymaktadır. Bu bölümde, KAP tabanlı tesis yerleşim problemleri için önerdiğimiz TZRE algoritmasına ayrıntılarıyla yer verilmiştir.

5.1. Çalışmanın Amacı

Bu çalışmanın başlıca motivasyonu, bölümler ve bölümleri oluşturan iş istasyonlarını, TZRE'deki birey yapısına uyarlayarak geliştirilecek algoritma ile herbir iş istasyonunun birey gibi davranmasını sağlamaktır. Böylelikle, bireyler (bölüm ya da iş istasyonları), arama alanını (yerleşim noktalarını) tarayıp bireysel en iyi konumlarını

ve diğer bireylerin (bölümlerin ya da iş istasyonlarının) o ana kadar bulundukları en iyi konumlarını hatırlayarak en az maliyetli yerleşimi belirleyeceklerdir.



Şekil 5.2 İş istasyonlarının TZRE ile yerleşim noktalarına atanmasının gösterimi

Bu durumda, bölümlerin şekli ile ilgili dikdörtgen, kare gibi varsayımlar yapılmaksızın; iş istasyonları, ilgili yerleşim noktalarına -bireysel ve bütünsel arama sonucu- yakınsayacak, bölümlerin şekilleri de bölümleri oluşturan iş istasyonlarının yerleşimi ile kendiliğinden oluşacaktır. Bu durumda bölümlerin şekilleri; en az maliyeti karşılayacak yerleşim şekline uygun olarak, dörtgen şekiller yanında, uygulamada sık karşılaşılmayan şekillerle kendiliğinden oluşabilecektir (Şekil-5.2).

Özetle; çalışmanın başlangıç noktasını, TZRE'nin temel mantığından hareketle, iş istasyonları yada bölümlerin (bireylerin) arama alanında (yerleşim noktaları) uçarak en iyi konumlarını (en az maliyetli yerleşim) bulmaları düşüncesi oluşturmaktadır.

Bu düşünceden hareketle, tesis yerleşim problemleri için TZRE algoritması geliştirme adımına geçilmiştir.

5.2. Algoritmanın Tanıtımı

Yerel arama teknikleri (bireylerin kendi deneyimleri) ile bütünsel arama tekniklerini (komşu bireylerin deneyimleri), keşfetme (exploration) ve yararlanmayı (exploitation) dengeleyerek birleştirme mantığıyla ortaya çıkan TZRE nin genel yapısı, Bölüm 4'te

ayrıntılı olarak açıklanmıştır. Bu alt bölümde, KAP tabanlı tesis yerleşim problemleri için önerilen TZRE algoritmasının adımları verilecektir.

Öncelikle, algoritmada kullanılacak birey yapısı Şekil 5.3'te gösterildiği şekilde tasarlanmıştır:

3	2	1	5	6	4

Şekil 5.3 Birey gösterimi

Şekil 5.3, bütün olarak TZRE[i]'yi, yani, i. bireyin konumunu göstermektedir. TZRE[i] vektörünün elemanları, diğer bir deyişle TZRE [i][j]'ler, i. bireydeki j. yerleşim bölgesine atanan bölüm sayılarıdır. Şekle göre, 1. yerleşim noktasına 3. bölüm atanmışken; 5. yerleşim noktasına 4. bölüm atanmıştır. Bu durumda, TZRE[i][1] = 3 iken TZRE[i][5] = 4'tür.

Algoritmanın tanıtılmasında kullanılan simgeler şunlardır:

- N: Bölüm/yerleşim bölgesi sayısı
- P: sürü büyüklüğü (sürü içindeki birey sayısı)
- TZRE[i]: (1...N) kümesinin değerlerini alan, N-boyutlu vektör olarak gösterilen TZRE populasyonunun i.bireyinin konumu
- TZRE[i][j]: i. bireyin j. yerleşim bölgesine atanan bölüm numarası
- Uyum[i]: i. bireyin maliyet (amaç) fonksiyonu

$$\min z = \sum_{i}^{n}\sum_{j}^{n}\sum_{k}^{n}\sum_{p}^{n} A_{ijkp} X_{ij} X_{kp}$$

$$A_{ijkp} = f_{ik} d_{jp}$$

f_{ik}= i. bölümden k. bölüme olan akış miktarı

d_{jp}= j. yerleşim noktasının p.yerleşim noktasına uzaklığı

X_{ij} = i. departman j.yerleşim noktasına atandığında 1; diğer durumlarda 0 değerini alır

X_{kp} = k. departman p.yerleşim noktasına atandığında 1; diğer durumlarda 0 değerini alır

- V[i][j]: i. bireyin j. yerleşim noktasının hızı (yol-mesafe)- N boyutlu vektör olarak kodlanır
- G_{best} = Bütünsel en iyi konum
- P_{best} [i] = i. bireyin yerel en iyi konumu
- P_{best}_uyum[i] = i. bireyin bulunduğu konumlar arasındaki en iyi yerel konumun uyum değeri

KAP tabanlı tesis yerleşim problemleri için önerilen TZRE algoritmasının adımları aşağıda verilmektedir:

1. Adım: Başlangıç

- For i = 1 to P (Sürüdeki tüm i.bireyler için)
    - Tüm j. yerleşim bölgeleri için
        - TZRE[i][j] 'yi, [1..N] kümesinden rassal olarak ata
        - V[i][j] 'yi oluştur
            - Eşitlik (4.4) ve Eşitlik (4.2) formüllerine göre belirle
        - uyum[i] yi hesapla
        - Bireyler arasında en iyi uyum değerine sahip bireyin indeksini G_{best} e ata
        - TZRE[i] nin değerlerini P_{best}[i] 'ye kopyalayarak ilk değerleri belirle
        - (4.4) ve (4.1) formüllerine göre V[i][j] ve TZRE[i][j] yi güncelle

2. Adım: Durma ölçütüne kadar tekrarla

- uyum[Gbest] ≥ uyum[i] olacak şekilde G_{best} 'i ata, $\forall i \leq P$
- Her birey için,
  - Pbest[i] = TZRE[i] sadece ve sadece uyum[i] > Pbest_fitness[i]
- Her birey i için,
  - Her yerleşim bölgesi j için,
    - Eşitlik (4.4) ve Eşitlik (4.1) formüllerine göre V[i][j] ve TZRE[i][j] yi güncelle

Algoritmada sözü edilen uyum değeri (uyum[i]) ,Bölüm 3.2.1'de açıklanan, Koopmans ve Beckmann tarafından geliştirilmiş KAP'ın tamsayılı doğrusal programlama modelinde kullanılan amaç fonksiyonu kullanılarak hesaplanmaktadır. Tesis yerleşim problemleri için geliştirilen TZRE algoritmasının adımları Şekil 5.4'te özetlenmiştir.

Adım 1: Başlangıç

- Pop[i][j] yi [1..N] kümesinden rassal olarak ata
- $\vec{v}_i$ yi, [-Vmax, Vmax] değerleri arasında rassal olarak ata
- KAP'ın amaç fonksiyonuna gore uyum[i]'yi hesapla
- En iyi uyum değerine sahip (en düşük maliyetli) bireyi $\vec{p}_g$ 'ye ata
- Pop[i]'yi kopyalayarak $\vec{p}_i$ 'yi oluştur
- $\vec{v}_i$ ve Pop[i][j]'yi aşağıdaki hız ve konum denklemlerine gore güncelle

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)) \quad (1)$$

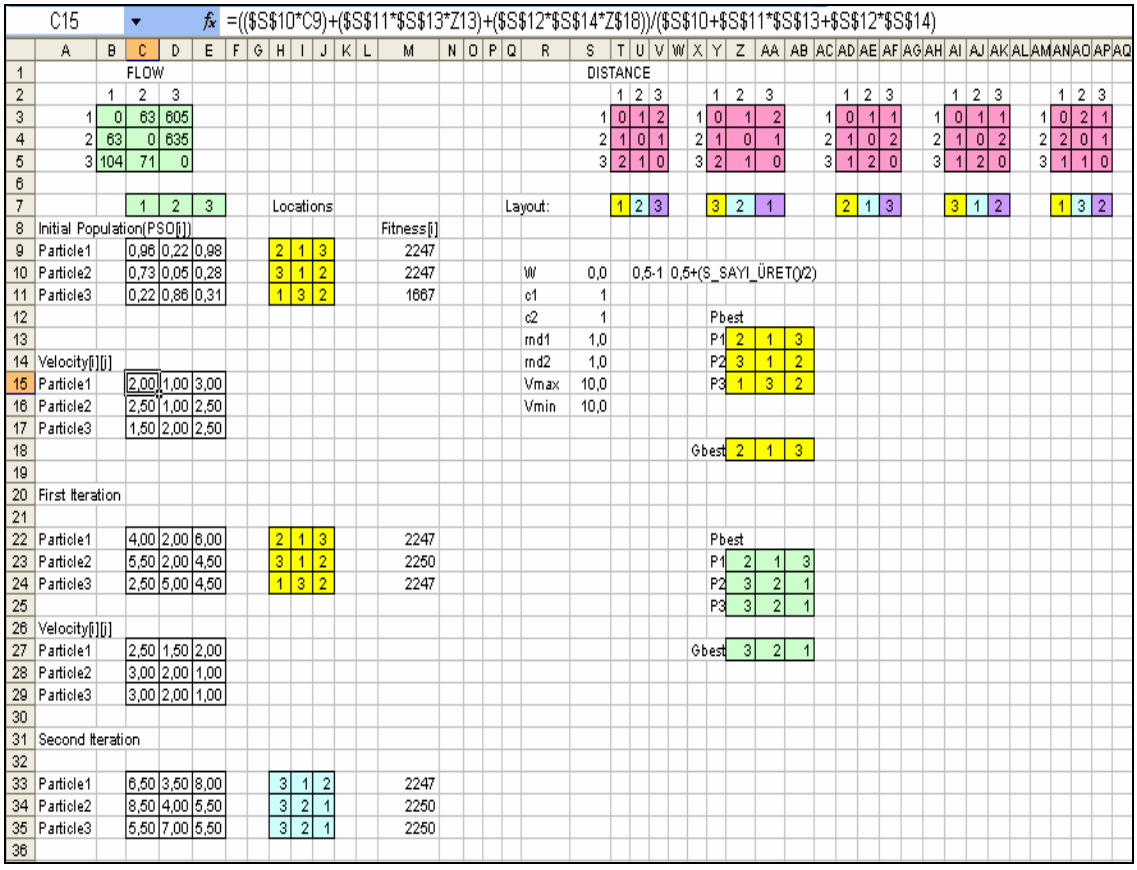
$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2)$$

Adım 2: Durma ölçütüne kadar tekrarla

- Find uyum [Gbest] ≥ uyum [i] olacak şekilde $\vec{v}_i$ 'yi hesapla
- $\vec{p}_i$ = Pop[i] eğer uyum [i] > p_uyum[i]
- Her birey I için;
- Her yerleşim noktası j için,
- $\vec{v}_i$ ve Pop[i][j]'yi Eşitlik (1) ve Eşitlik (2)'ye göre güncelle

Şekil 5.4 Takım Zekası Tabanlı Rassal Eniyileme Algoritması (TZRE) adımları

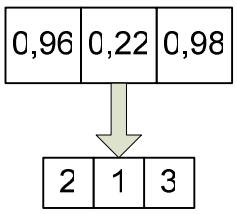
Önerilen algoritmanın çalıştırılması için geniş çaplı bir yazılım geliştirilmeden önce, algoritmanın daha iyi anlaşılabilmesi için MS Excel ortamında algoritma adımlarının benzetimi, başlangıç olarak 3 bölüm için tasarlanmıştır (Şekil 5.5):



Şekil 5.5 Taslak TZRE algoritması

Algoritma adımlarının benzetimine, akış ve uzaklık matrisleri belli olan 3 bölüm için başlangıç populasyonu oluşturularak başlanmıştır. Başlangıç populasyonu, "rassal anahtar (random key)" kodlaması ile belirlenmiştir.

Rassal anahtar kodlamasında öncelikle, her TZRE[i][j] (PSO[i][j]), diğer bir deyişle i. bireydeki j.yerleşim bölgesine atanan bölüm sayısı, [0,1] aralığında rassal olarak atanmış (Şekil 5.5'de "Initial Population" satırının altındaki populasyon gösterimi); atanan rassal sayılar küçükten büyüğe doğru sıralanarak yerleşim bölgeleriyle eşleşen bölüm sayıları belirlenmiştir (Şekil 5.6).



Şekil 5.6 Rassal anahtar kodlaması

Başlangıç populasyonu oluşturulduktan sonra, herbir bireyin hızları (V[i][j]); Eşitlik (4.4) ve Eşitlik (4.3) ile atalet ağırlığı (w), hızlandırma katsayıları (c_1 ve c_2) , maksimum ve minimum hız parametreleri kullanılarak hesaplanmıştır. İlk iterasyonda, Eşitlik (4.2) kullanılarak bireylerin yeni konumları hesaplanmış; rassal anahtar kodlaması ile bölümler yerleşim noktalarına atanmış; herbir bireyin uyum değerleri hesaplanmıştır. Ayrıca bireylerin bireysel en iyi uyum değerleri (P_{best}) ve bütünsen el en iyi uyum değerleri (G_{best}), Şekil 5.5'te görüldüğü gibi ayrıca tutulmaktadır. İkinci iterasyonun da benzer şekilde benzetimi yapılarak en düşük uyum değerine sahip birey (en az maliyetli yerleşim) elde edilmiştir.

Bu taslak, algoritmada geçen hız ve konum kavramları, bireylerin hızlarının Eşitlik (4.2), Eşitlik (4.3) ve Eşitlik (4.4) formüllerine göre hesaplanış yönteminin belirlenmesi ve algoritma parametrelerinin daha iyi anlaşılması açısından yarar sağlamaktadır.

Bu çalışmada, TZRE algoritmasının başarısını analiz edebilmek amacıyla 3 farklı algoritma geliştirilmiştir.

TZRE-St (Standart TZRE): Adımları bu alt bölümde ayrıntılı olarak açıklanan ve Şekil 5-4'te verilen standart TZRE algoritmasıdır.

TZRE-VR (Velocity Restricted- Hız Kısıtlı- TZRE): TZRE-St'nin, birey hızlarının ulaşabileceği en küçük ve en büyük değerlerinin, kullanıcı tarafından [$-V_{max}$, V_{max}] aralığında belirlendiği halidir. Standart TZRE (TZRE-St) algoritmasından farkı,

TZRE-St'de bireylerin hızları (V[i][j]); Eşitlik (4.4) ve Eşitlik (4.2) formüllerine göre belirlenirken,

TZRE-VR'de bireylerin hızları (V[i][j]); [$-V_{max}$, V_{max}] arasında rassal olarak belirlenmektedir.

V_{max}'ın tariflenmesi, sürü bir noktaya yakınsama ya da bozulma eğilimi göstermese de, bireylerin sınırlar arasında salınımına izin verir. Sürü, yakınsama olmadan, salınımlarıyla, optimum alanda iyi noktalara (çözümlere) ulaşır.

TZRE-M (Modified TZRE): TZRE-St ve TZRE-VR algoritmaları ile çözülen problemler analiz edildiğinde, çözümlerin sıklıkla yerel en iyiye yakınsadığı görülmüştür. Erken yakınsamayı önleyerek daha kaliteli çözümler elde etmek amacıyla, kullanıcının belirlediği (Şekil 5.7'de Stop Iter. Bölümü) peryotta, populasyonun tekrar oluşturulduğu veya atalet ağırlığı *(w)*'nin [0,4; 0.9] aralığında rassal olarak değişebildiği TZRE-M algoritması geliştirilmiştir. Kullanıcı peryodik olarak populasyon oluşturma ve atalet ağırlığı değiştirme seçeneklerinden birini tercih etmelidir.

5.3. Kontrol Parametrelerinin Seçimi

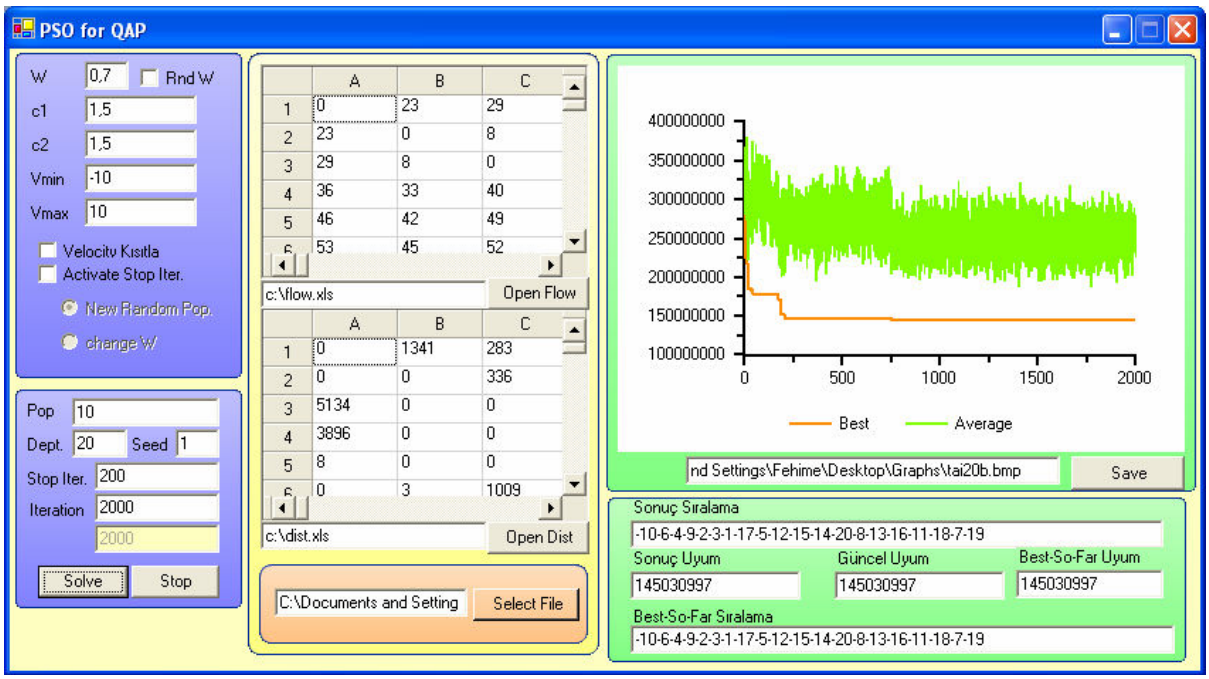
TZRE algoritmasının KAP alanına uygulandığı bu deneysel çalışmada, TZRE ile ilgili önceden yapılan mevcut araştırmalar sonucu elde edilmiş kontrol parametreleri kullanılmıştır. TZRE kontrol parametreleri; hızlandırma katsayıları *(c_1 ve c_2)*, atalet ağırlığı *(w)*, hız kısıtı *(v_{max})* ve sürü büyüklüğü *(p)* dür (Bergh and Engelbrecht, 2005). TZRE'nin kontrol parametrelerinin seçimi ile ilgili yapılmış çalışmalar incelendiğinde, Shi ve Eberhart (1998), $c_j = 2.0, w_{\max} = 0.9, w_{\min} = 0.4$ parametre değerlerinin kullanılabileceğini belirtmişlerdir.

Ayrıca, Bergh and Engelbrecht (2005), atalet ağırlığı ve hızlandırma katsayıları arasındaki genel ilişkiyi, $w > \frac{1}{2}(c_1 + c_2) - 1$ şeklinde karakterize etmiş ve atalet ağırlığı için 0.7298; hızlandırma katsayıları için 1.49618 değerlerini önermiştir.

Yapılan deneysel çalışmalarda kullanılan parametre değerlerine, ilgili çalışmalar için yapılan açıklamalarda değinilmiştir.

5.4. Deneysel Sonuçlar

KAP tabanlı tesis yerleşim problemleri için önerilen TZRE algoritması için C# kullanılarak bir program yazılmış ve tüm deneysel çalışmalar bu ortamda gerçekleştirilmiştir (EK-2).



Şekil 5.7 Önerilen TZRE algoritması için geliştirilen program arayüzü

Program girdileri, bölümler arası akış ve uzaklık matrisleridir. Matrisler ilgili Excel dosyaları (flow.xls ve distance.xls) seçilerek programa çağrılmakta; TZRE'nin Bölüm 5.2'de açıklanan parametreleri kullanıcı tarafından girildikten sonra program istenen tekrar sayısına kadar çalıştırılmakta ve o ana kadar elde edilen en iyi bölüm sıralaması ve amaç fonksiyonu (maliyet) değeri ile son tekrarın bölüm sıralaması ve amaç fonksiyonu (maliyet) değeri, programın çıktısı olarak verilmektedir. Ayrıca, Şekil 5.7'de, önerilen TZRE algoritması için geliştirilen program arayüzünde görülen tekrar sayıları (iterasyon) ve uyum değerleri arasındaki ilişki aşağıda verilmiştir:

Güncel tekrar sayısı = Tekrar sayısı ise

O ana kadar en iyi (Best So-Far) uyum değeri = Sonuç uyum değeri

Program, *TZRE-St*, *TZRE-VR* ve *TZRE-M* olarak geliştirilen 3 farklı TZRE algoritması için çalıştırılabilmektedir.

Ayrıca, TZRE'nin başarısını daha ayrıntılı izleyebilmek amacıyla program çalıştırıldığı süre içerisinde ilgili problem için, en iyi ve ortalama uyum değerleri iterasyon bazında grafik olarak tutulmaktadır. Bu grafik yakınsama analizi gibi çalışmalar için kullanıcı tarafından istenilen yere kaydedilebilmektedir.

5.4.1. İlk problem seti deneyleri ve sonuçları

İlk problem seti, Golany ve Rosenblatt (1989)'ın KAP tabanlı tesis yerleşim problemleri ile ilgili yaptığı çalışmada kullandıkları test problemleridir. Bu problemler, tesis yerleşim problemleri ile ilgili birçok çalışmada kıyaslama amacıyla kullanılmaktadır. Bu ilk set, TZRE 'nin başarısını ortaya çıkartmak amacıyla kullanılmıştır.

Program öncelikle, TZRE ile ilgili incelenen çalışmalarda en sık rastlanan $c_1 = c_2 = 2.0, w = 0.9$ parametre seti kullanılarak (Shi ve Eberhart, 1998), *TZRE-St* algoritması ile herbir problem için 100 tekrar sayısıyla çalıştırılmıştır.

Tablo 5-1 TZRE algoritmasının ilk deneysel sonuçları

Problem	TZRE Çözümü	En İyi Çözüm	Yakınsama Oranı (%)
Rosenblatt (1)	12822	12822	100
Rosenblatt (2)	14853	14853	100
Rosenblatt (3)	13172	13172	100
Rosenblatt (4)	13032	13032	100
Rosenblatt (5)	12819	12819	100
Zimmermann and Sovereign	389	389	100
Gavett	403	403	100

İlk problem setinin sonuçları, Tablo 5.1'de verilmiştir. Maksimum iterasyon sayısı 100 gibi düşük bir sayıyla sınırlandırılmasına rağmen, TZRE çözümü tüm problemler için bilinen en iyi çözüme ulaşmıştır. Sonuçlarla ilgili ayrıntılı bilgi EK-3'te verilmiştir

5.4.2. QAPLIB problemleri için yapılan deneysel çalışma ve sonuçları

İlk deney setinin sonuçlarının her problemde %100 eniyilik seviyesine ulaşması, deneysel çalışmaları daha ileriye götürerek; geliştirilen TZRE algoritmasının, çok sayıda KAP problem örneklerini içeren Burkard (1996)'ın QAPLIB (QAP Library) problemleriyle test edilmesine teşvik etmiştir.

Bir sonraki adım olarak, QAPLIB'den alınan 30 bölüm ve altındaki bütün problemlerle çalışılmıştır. Bu set, 77 adet problemden oluşmaktadır. Her problem, Bölüm 5.1'te özellikleri açıklanan *TZRE-St*, *TZRE-VR* ve *TZRE-M* olarak adlandırılmış 3 farklı TZRE algoritması ile çözülmüştür.

Bu deney seti için, farklı problem türleri ve farklı parametre değerleriyle yapılan çalışmalarla, TZRE ile ilgili literatürdeki çalışmalarda sık karşılaşılan değerler arasından $c_1 = c_2 = 1.0, w_{\min} = 0.4, w_{\max} = 0{,}9$ ile TZRE-M için, Bergh ve Engelbrecht (2005)'in çalışmalarında kullandıkları $c_1 = c_2 = 1.49618, w = 0.7298$ parametre değerleri seçilmiştir. 77 problem için, üç farklı algoritmanın çalıştırılması sonucunda elde edilen 231 adet sonuç, EK-4'te verilmiştir.

En iyi çözümleri bilinen problemlerin en alt satırda "En iyi çözmü değeri" başlığı ile çözümleri verilmiş; diğer problemlerin de GRASP (Greedy Randomized Adaptive Search Procedure) adı verilen üstsezgiselle bulunmuş bilinen iyi çözümleri QAPLIB'den alınarak tabloya eklenmişir. TZRE-St, TZRE-VR ve TZRE-M algoritmaları ile çözülen 77 adet problemden bir kısmının çözümleri Tablo 5-2'de verilmiştir. Örneğin, ilk problem Bur26a, 26 bölümlü bir KAP problemidir. Bur26a'nın en iyi çözüm değeri henüz bulunamamış; bilinen en iyi çözümü GRASP (Greedy Randomized Adaptive Search) üstsezgiseli ile 5426670 olarak elde edilmiştir.Bur26a problemi, geliştirilen TZRE-St, TZRE-VR ve TZRE-M algoritmaları ile bölüm sayısının 100 katı kadat tekrar sayısı ile çözülmüş; elde edilen sonuçlar GRASP değeri ile karşılaştırılarak herbir algoritma için bilinen en iyi çözüme yakınsama yüzdeleri belirlenmiştir (Tablo 5-2).

Tablo 5-2 QAPLIB problemleri için deneysel sonuçlar

No	Problem	Çözüm Yöntemi	Uyum Değeri	Opt. (%)	No	Problem	Çözüm Yöntemi	Uyum Değeri	Opt. (%)
1	Bur26a	TZRE-St	5549427	98	9	Esc16b	TZRE-St	292	100
		TZRE-VR	5577499	97			TZRE-VR	292	100
		TZRE-M	5543157	98			TZRE-M	292	100
		En iyi çözüm	5426670				**En iyi çözüm**	292	
2	Bur26b	TZRE-St	3925079	97	10	Esc16h	TZRE-St	1006	99
		TZRE-VR	3901073	98			TZRE-VR	1004	99
		TZRE-M	3908419	98			TZRE-M	996	100
		Grasp	3817852				**Grasp**	996	
3	Bur26c	TZRE-St	5574281	97	11	Esc16i	TZRE-St	14	100
		TZRE-VR	5615247	97			TZRE-VR	16	88
		TZRE-M	5583668	97			TZRE-M	14	100
		Grasp	5426795				**Grasp**	14	
4	Bur26d	TZRE-St	3925173	97	12	Lipa30a	TZRE-St	13620	97
		TZRE-VR	3939160	97			TZRE-VR	13620	97
		TZRE-M	3938374	97			TZRE-M	13620	97
		Grasp	3821225				**Grasp**	13178	
5	Bur26e	TZRE-St	5535373	97	13	Lipa20a	TZRE-St	3828	96
		TZRE-VR	5524089	98			TZRE-VR	3828	96
		TZRE-M	5568921	97			TZRE-M	3828	96
		Grasp	5386879				**Grasp**	3683	
6	Bur26f	TZRE-St	3882849	97	14	Had18	TZRE-St	5572	96
		TZRE-VR	3915030	97			TZRE-VR	5542	97
		TZRE-M	3901009	97			TZRE-M	5472	98
		Grasp	3782044				**Grasp**	5358	
7	Bur26g	TZRE-St	10363373	98	15	Tai15b	TZRE-St	52385462	99
		TZRE-VR	10406493	97			TZRE-VR	52349005	99
		TZRE-M	10323363	98			TZRE-M	52435041	99
		Grasp	10117172				**Grasp**	51765268	
8	Bur26h	TZRE-St	7281467	97	16	Had12	TZRE-St	1692	98
		TZRE-VR	7311706	97			TZRE-VR	1686	98
		TZRE-M	7281467	97			TZRE-M	1682	98
		Grasp	7098658				**En iyi çözüm**	1652	

TZRE algoritmalarının başarılarını karşılaştırmada, yakınsama yüzdeleri (Tablo 5-2'de Opt(%) sütunu) kullanılmıştır:

$$Bilinen\ \ en\ \ iyi\ \ çözüme\ \ yakinsama\ \ yüzdesi = \frac{Bilinen\ \ en\ \ iyi\ \ çözüm}{TZRE\ \ çözümü} * 100$$

Örneğin, Bur26a problemi için, TZRE-St'nin bilinen en iyi çözüme yakınsama yüzdesi; 5426670/ 5549427= 98 olarak hesaplanmıştır.

Üç farklı TZRE algoritmasının çalıştırılması sonucunda, 77 problem için problemlerin %83 'ünün bilinen en iyi çözüme %100 yakınsadığı görülmüştür.

Bu sonuç, kontrol parametreleri için genel kabul görmüş değerlerin kullanıldığı ve probleme özel ayarlamaların yapılmadığı göz önüne alınarak değerlendirilmelidir. Bu 77 problemin üç farklı algoritma kullanılarak çalıştırılması sonucu elde edilen 231 sonucun 62 sinde en iyi / iyi çözüme yaklaşma oranı %95 ve üzeridir (Tablo 5-3).

Tablo 5-3 QAPLIB problemleri için özet sonuçlar

Bilinen en iyi çözüm değerine yakınsama oranı	Sıklık
95-100	62
80-94	107
60-79	34
59 or less	28
Toplam	231

Her bir problem bireysel olarak ele alınıp probleme özel parametre ayarlaması yapıldığında, bu sonuçlarda gözle görülür derecede artış olacağını söylemek mümkündür.

QAPLIB'den alınan 77 problem üzerinde, geliştirilen 3 algoritmanın başarısı, öncelikle problem büyüklüklerine (bölüm sayıları) göre ayrıntılı olarak karşılaştırılmıştır (EK-5). Tablo 5-4'te, sözü edilen karşılaştırmanın özet bilgileri yer almaktadır.

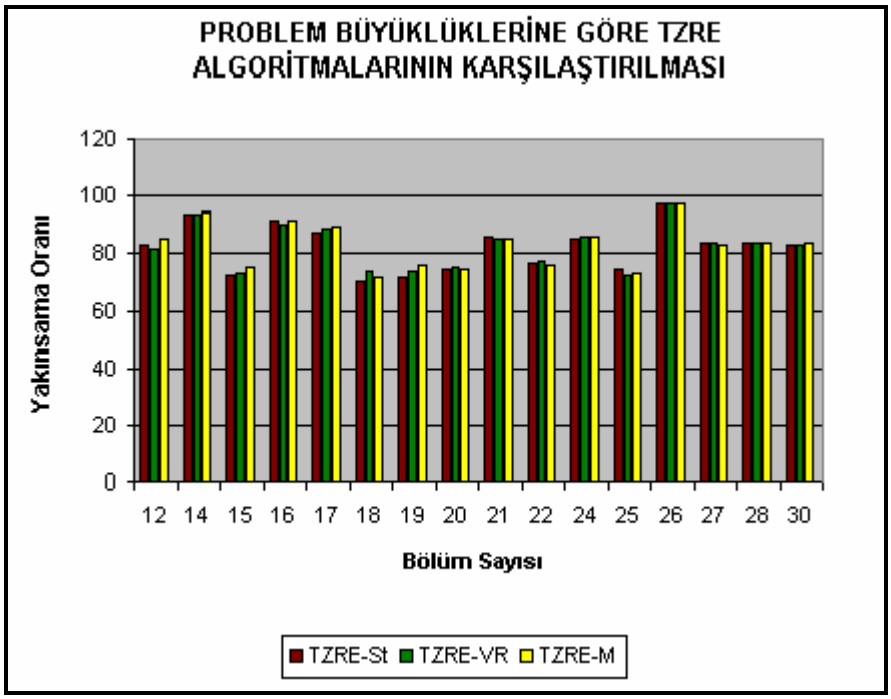
Bu bilgilere göre, bilinen en iyi çözüme yakınsama oranı 26 bölümlü problem setinde en yüksek çıkmış; 26 bölümlü 8 problemin 3 farklı TZRE algoritması ile çözümünün sonuçlarının ortalaması alınarak %97 olarak hesaplanmıştır. Bilinen en iyi çözüme yakınsama oranının en düşük değeri olan %72, 18 bölümlü 4 problemin yakınsama ortalamaları alınarak elde edilmiştir.

Tablo 5-4 Problem büyüklükleriyle yakınsama oranlarının karşılaştırılması

Problem Büyüklüğü	Problem Sayısı	Yakınsama Oranı (%)			
		TZRE-St	TZRE-VR	TZRE-M	Ort.
12	9	83	82	85	83
14	2	94	94	95	94
15	8	73	73	75	74
16	13	92	90	92	91
17	2	88	89	90	89
18	4	71	74	72	72
19	1	72	74	76	74
20	11	74	75	75	75
21	1	86	85	85	85
22	3	77	77	76	77
24	1	85	86	86	86
25	4	75	72	73	73
26	8	97	97	97	97
27	1	84	84	83	84
28	1	84	84	84	84
30	8	83	83	84	83

Problem setlerinin, bölüm sayılarına göre, bilinen eniyi çözüme ortalama yakınsama oranlarının büyükten küçüğe sıralaması; 26- 14- 16- 17- 24- 21-27- 28- 12- 30- 22- 20- 15- 19- 25- 18 şeklindedir. TZRE algoritmasının başarısı; bölüm sayısı fazla olan problemlerde azalmamıştır. En düşük bölüm sayısına sahip 12 bölümlü problem setlerinde yakınsama oranı %83 iken en yüksek bölüm sayısına sahip 30 bölümlü sette yakınsama oranı %83 olarak kalmıştır.

Özetle, EK-5 ve Tablo 5-4'teki bilgiler doğrultusunda, TZRE algoritmasının başarısının problem büyüklüğüne (bölüm sayısına) bağlı olmadığı görülmektedir. 26 bölümlü problemlerde dahi bilinen en iyi çözüme yakınsama oranı %97 gibi yüksek bir değer olarak elde edilmiştir.



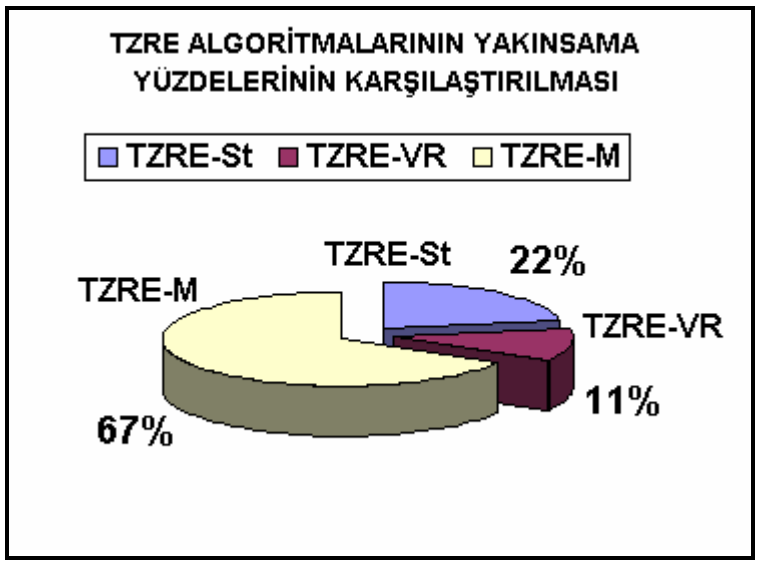
Şekil 5.8 Problem büyüklüklerine göre TZRE algoritmalarının karşılaştırılması

TZRE-St ve TZRE-VR algoritmaları ile çözülen problemler analiz edildiğinde, çözümlerin sıklıkla yerel en iyiye yakınsadığı görülmüştür.. Erken yakınsamayı önleyerek daha kaliteli çözümler elde etmek amacıyla, kullanıcının belirlediği peryotta populasyonun tekrar oluşturulduğu veya atalet ağırlığı *(w)*'nin belli bir aralıkta rassal olarak değişebildiği TZRE-M algoritması geliştirilmiştir.

Geliştirilen TZRE-M algoritmasının başarısını değerlendirmek ve TZRE-St ve TZRE-VR algoritmalarıyla karşılaştırmak için yine EK-5 ve Tablo 5-4'ten yararlanılmıştır. QAPLIB 'ten alınan 77 adet problem, bölüm sayılarına göre sınıflandırıldığında 12 bölümden 30 bölüme kadar toplam 16 adet problem seti elde edilmektedir. 16 adet problem seti içerisinde 12, 14, 15, 17, 19, 30 bölüm sayısına sahip olan 6 problem setinde TZRE-M algoritmasıyla, TZRE-St ve TZRE-VR'ye göre daha iyi sonuçlar elde edilmiştir. 16 adet problem seti içerisinde, 21 ve 25 bölüm sayılı problem setlerinde TZRE-St ile elde edilen, bilinen en iyi çözüme yakınsama oranı; diğer iki algoritmaya göre daha yüksektir. 16 adet problem seti içerisinde sadece 18

bölümlü problemde TZRE-VR ile diğer iki algoritmaya göre daha iyi çözüm elde edilmiştir.

TZRE algoritmalarının problem büyüklüklerine göre bilinen en iyi çözüme yakınsama oranlarının karşılaştırılması sonucunda elde edilen grafik Şekil 5-9'da verilmektedir.



Şekil 5.9 TZRE algoritmalarının yakınsama yüzdelerinin karşılaştırılması

TZRE-St ve TZRE-VR ile elde edilen çözümlerin kalitesini daha da yükseltmek amacıyla geliştirilen TZRE-M algoritması, problem setlerinin %67'sinde diğer iki algoritmaya göre daha iyi sonuç vermiştir.

Bireylerin hızlarının sınırlandırılmasının TZRE algoritmasının başarısına etkisi incelendiğinde, ele alınan 77 problemin 24'ünde TZRE-VR algoritmasının kullanılması, en iyi çözüme yaklaşma yüzdesini arttırmış; 26 problemde herhangi bir değişiklik yapmamış; 27 problemde ise azaltmıştır. Bu sonuç, bireylerin hızın kısıtlanmasının bilinen en iyi çözüme yaklaşmadaki etkisinin problemin yapısına bağlı olarak değişkenlik gösterdiğini belirtmektedir.

5.4.3. Parametre Analizi

Parametre seçiminin TZRE'nin başarısı üzerindeki etkisini incelemek amacıyla, QAPLIB'den alınan 12, 15, 16, 20, 26 ve 30 bölüm sayılı problemler üzerinde farklı parametre değerleri ile çalışılarak sonuçlar yorumlanmıştır. Herbir problem için tekrar sayısı, bölüm sayısının yüz ile çarpımıyla belirlenmiştir. Diğer ayrıntılar Tablo 5.3'te verildiği gibidir.

Farklı bölüm sayılarına sahip 6 problem için yapılan parametre analizinde, Bölüm 5.3.1 ve Bölüm 5.3.2' de kullanılan parametre değerlerinin geçerliliği doğrulanmış ve ek olarak, belirlenen farklı parametre setleri için de elde edilen değerler yorumlanmıştır. Yapılan parametre ayarı, Chr12 gibi bazı problemlerde, en iyi çözüme yaklaşma yüzdesinin 57'den 86'ya çıkmasını sağlamıştır (Tablo 5-5). Bilinen en iyi çözüme yakınsama oranı %97-%100 gibi yüksek değerler arasında değişen Bur26, Tai15b gibi problemler için farklı parametre değerlerinin kullanılması algoritmanın performansını değiştirmemiştir.

Parametrelerin etkisini ortaya çıkartmak için seçilen aralıklar yeteri kadar geniş olmadığı için çözüm kalitesinde herhangi bir bozulmaya neden olmamıştır. Bu nedenle Tablo 5.5'te verilen aralıktaki parametre değerleri; eğer kuşak sayısı, parametre değerlerinin küçük aralıklarda değişmesi (ince ayar) ile büyük aralıklarda değişmesi (kaba ayar) arasındaki farkı karşılayacak kadar büyük tutulursa, rahatlıkla kullanılabilir.

5.5. Bölüme Özel Kısıtları Olan KAP Tabanlı Yerleşim Problemleri

Tesislerin yeniden yerleşim ya da yeniden düzenleme kararları için *"herhangi bir bölüm herhangi bir yerleşim bölgesine atanabilir"* varsayımı; zemin özellikleri , aydınlatma, titreşim ve gürültü, güvenlikle ilgili düzenlemeler ile sütun ve kiriş yapıları gibi bina özellikleriyle ilgili kısıtlar nedeniyle her zaman geçerli değildir. Bazen, tahmin edilen alan gereksinimi; etkili ve kullanılabilir alana uymayabilir. Bu nedenle, bazı bölümler, bu tür gereksinimleri karşılamak amacıyla, sadece belli yerleşim noktalarına atanabilirler.

Tablo 5-5 Farklı parametre değerleri için deneysel sonuçlar

Parametre Değerleri	Problem	Uyum değeri	Opt. (%)	Problem	Uyum değeri	Opt. (%)
W=0,9 c1=c2=2	Bur26a	5549427	98	Esc16b	292	100
W=0,9 c1=c2=1 VR		5568439	97		292	100
W=0,9 c1=c2=1,4 VR		5582947	97		292	100
W=0,9 c1=c2=2 VR		5577499	97		292	100
TZRE-M		5543157	98		292	100
W=0,4 c1=c2=2 VR		5552878	98		292	100
W=0,5 c1=c2=2 VR		5556804	98		292	100
W=0,6 c1=c2=2 VR		5565628	98		292	100
W=0,7 c1=c2=2 VR		5555921	98		292	100
W=0,7298 c1=c2=1,49618		5540542	98		292	100
W=0,8 c1=c2=2 VR		5553406	98		292	100
W=0,9 c1=c2=2 VR		5550541	98		292	100
W=1 c1=c2=2 VR		5569195	97		292	100
En iyi çözüm değeri		5426670			292	
W=0,9 c1=c2=2	Had20	7182	96	Lipa30a	13620	97
W=0,9 c1=c2=1 VR		7290	95		13631	97
W=0,9 c1=c2=1,4 VR		7290	95		13644	97
W=0,9 c1=c2=2 VR		7172	97		13653	97
TZRE-M		7150	97		13620	97
W=0,4 c1=c2=2 VR		7272	95		13606	97
W=0,5 c1=c2=2 VR		7262	95		13633	97
W=0,6 c1=c2=2 VR		7216	96		13649	97
W=0,7 c1=c2=2 VR		7266	95		13616	97
W=0,7298 c1=c2=1,49618		7196	96		13613	97
W=0,8 c1=c2=2 VR		7222	96		13643	97
W=0,9 c1=c2=2 VR		7310	95		13643	97
W=1 c1=c2=2 VR		7116	97		13640	97
En iyi çözüm değeri		6922			13178	
W=0,9 c1=c2=2	Tai15b	52385462	99	Chr12b	16996	57
W=0,9 c1=c2=1 VR		52418492	99		15146	64
W=0,9 c1=c2=1,4 VR		52410710	99		14154	69
W=0,9 c1=c2=2 VR		52349005	99		16996	57
TZRE-M		52435041	99		12432	78
W=0,4 c1=c2=2 VR		52362487	99		14398	68
W=0,5 c1=c2=2 VR		52356719	99		15520	63
W=0,6 c1=c2=2 VR		52613739	98		13312	73
W=0,7 c1=c2=2 VR		52586955	98		12324	79
W=0,7298 c1=c2=1,49618		52498707	99		11490	85
W=0,8 c1=c2=2 VR		52404105	99		11308	86
W=0,9 c1=c2=2 VR		52498707	99		15814	62
W=1 c1=c2=2 VR		52482878	99		14616	67
En iyi çözüm değeri		51765268			9742	

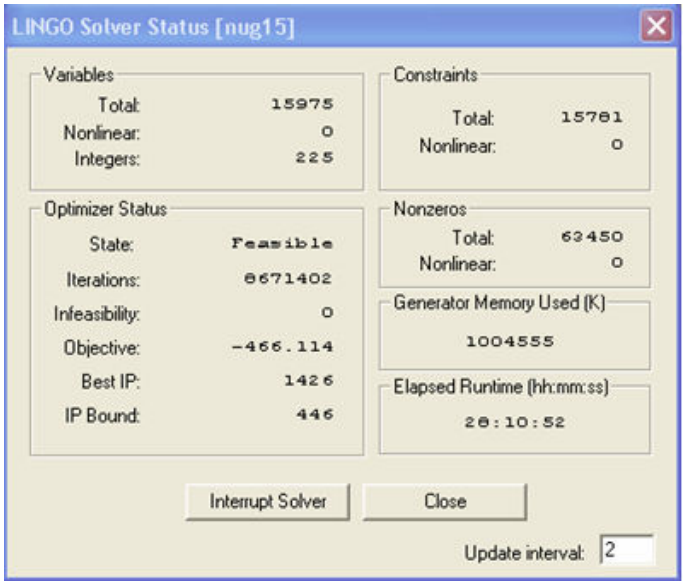
Bu kısıtları KAP tabanlı tesis yerleşim problemlerine yansıtabilmek için farklı model türleri kullanılabilir. Örneğin, KAP modelinde, istenmeyen atamaları temsil eden karar değişkenlerini kaldırmak, bir çözüm olarak uygulanabilir. Bir diğer yaklaşım, formülasyonda herhangi bir değişiklik yapmadan istenmeyen bir atamayı ortadan kaldırmak için ceza (penalty) uygulanması olabilir. Örneğin, M yeteri kadar büyük pozitif bir sayı olmak üzere, amaç fonksiyonunda ilgili terimlere M değeri eklenebilir. Bu tür KAP modeli seçenekleri, gerçek hayattaki uygulamalar için rahatlıkla kullanılabilirler.

Problemi daha gerçekçi olarak yansıtabilmek için, bazı varsayımların gevşetilmesi ve değiştirilmesine ek olarak, eşit olmayan boyutlara sahip bölümlerin yerleşimlerini belirlemek için de KAP modelleri formülasyonları kullanılabilir.

Bölüme özel kısıtların, önerilen TZRE tabanlı algoritmanın başarısına etkisini incelemek amacıyla, QAPLIB'den alınan ve önceki bölümde ayrıntılı olarak değinilen problemlerden bir kısmı seçilmiştir.

Seçilen problemlere, %10, %20, %30 olmak üzere 3 farklı düzeyde kısıt eklenmiştir. Belli yerleşim bölgelerine istenmeyen bölüm atamalarının yapılmaması, problemlere bu şekilde yansıtılmıştır. Kısıt düzeyleri, rassal olarak seçilmiş, atama yapılamayan bölüm ve yerleşim bölgesi çiftlerinin, toplam bölüm sayısına oranını göstermektedir. Örneğin, Had20 problemi için %20 kısıt düzeyi, 20 bölüm içerisinde 4 bölümün belirli dört yerleşim noktasına atanamayacağı anlamına gelmektedir. Bir başka deyişle Had20 problemi için kısıt düzeyinin %20 olması, 20 bölüm içerisinde, örneğin, 3.bölümün 5. yerleşim noktasına; 18.bölümün 12. yerleşim noktasına; 9. bölümün 7.yerleşim noktasına ve 16. bölümün 10. yerleşim noktasına atanamayacağı gibi 4 bölüm-yerleşim çifti kısıtını ifade eder.

Bu amaçla, QAPLIB'den alınan 6 problem; %10, %20 ve %30 düzeylerinde rassal kısıtlar eklendikten sonra Bölüm 5.2'de açıklanan TZRE-M yaklaşımı ile çözülmüştür.



Şekil 5.10 Örnek Lingo sonuç ekranı

Rassal olarak türetilen kısıtların eklendiği problemlere, önerilen algoritma ile çözüm aranmadan önce, elde edilen sonuçlarla ilgili karşılaştırma yapılabilmesi için LINGO ile bu 6 problem için en iyi çözüm araştırılmıştır. Problemlerin bir kısmı için makul sürede eniyi çözüm bulunamamıştır. Bu nedenle, o ana kadar elde edilen en iyi çözümlerle önerilen algoritma karşılaştırılmıştır. Örneğin, Şekil. 5.10'da, 15 bölümlü Nug15 problemi için çalıştırılan LINGO'nun 28.6 saat sonraki sonuç ekranı görüntüsü verilmiştir. 25 bölümlü Nug25 probleminin LINGO'da çözümünden 82 sayfalık sonuç çıktısı elde edilmiş ve örnek olması açısından problem için kurulan model ve çıktısının ilk sayfası EK-6'da verilmiştir.

Bölüme özel kısıtlar içeren KAP tabanlı tesis yerleşim problemlerinin önerilen TZRE algoritması ile çözüm sonuçları Tablo 5.6'da verilmiştir. Tablo incelendiğinde, artan kısıt düzeyinin amaç fonksiyonunu arttırdığı görülmektedir. Kısıt düzeyinin artması, atama yapılamayacak bölüm-yerleşim noktası çifti sayısını arttırdığı için; bölüm kısıtı olan problemin çözümünün (amaç fonksiyonu, uyum değeri, maliyet değeri); aynı problemin atama kısıtı olmayan çözümünden uzaklaşmasını sağlaycaktır. Bu nedenle, kısıt düzeyi artışının uyum değerini arttırması beklenen bir durumdur.

Tablo 5-6 Bölüm kısıtı olan problemler için deneysel sonuçlar

Problem	Yöntem /Kısıt Düzeyi	Uyum Değeri	Opt. (%)	Problem	Yöntem /Kısıt Düzeyi	Uyum Değeri	Opt. (%)
Esc16a	TZRE-St	74	92	Had20	TZRE-St	7182	96
	TZRE-VR	74	92		TZRE-VR	7172	97
	TZRE-M	72	94		TZRE-M	7150	97
Kısıt Düzeyi	10%	70	97	Kısıt Düzeyi	10%	7164	97
	20%	70	97		20%	7186	96
	30%	86	79		30%	7270	95
	En iyi çözüm	68			**En iyi çözüm**	6922	
Esc16b	TZRE-St	292	100	Nug12	TZRE-St	620	93
	TZRE-VR	292	100		TZRE-VR	630	92
	TZRE-M	292	100		TZRE-M	630	92
Kısıt Düzeyi	10%	292	100	Kısıt Düzeyi	10%	612	94
	20%	292	100		20%	620	93
	30%	292	100		30%	620	93
	En iyi çözüm	292			**En iyi çözüm**	578	
Nug15	TZRE-St	1304	88	Nug25	TZRE-St	4356	86
	TZRE-VR	1294	89		TZRE-VR	4324	87
	TZRE-M	1266	91		TZRE-M	4230	89
Kısıt Düzeyi	10%	1208	95	Kısıt Düzeyi	10%	4256	88
	20%	1272	90		20%	4374	86
	30%	1304	88		30%	4544	82
	En iyi çözüm	1150			**En iyi çözüm**	3744	

SONUÇ ve ÖNERİLER

Modüler üretim teknolojilerinin sağladığı olanaklardan etken (effective) bir şekilde yararlanmak ancak yeniden yerleşim problemlerinin etkin (efficient) bir çözüm algoritmasına kavuşturulması ile olasıdır. Esnek ve modüler imalat teknolojileri, tesislerin mevcut yerleşiminin peryodik ve/veya anlık doğrulanması ihtiyacını doğurmaktadır. Ortaya çıkan bu ihtiyaçlar nedeniyle, tesis yerleşim problemlerini çözüm bulabilmek için kullanılan algoritmalar önem kazanmıştır.

Bu çalışmada, KAP tabanlı yerleşim problemleri için TZRE tabanlı bir algoritma önerilmiştir. TZRE, kuş ve balık sürüleri gibi organizmaların sosyal davranışları ve bir grup halinde hedeflere ulaşmada yaptıkları işbirliğinden esinlenerek bir optimizasyon aracı olarak son yıllarda özellikle kombinatoryal problemlerin çözümünde yaygın olarak kullanılmaya başlanmış; popülaritesi giderek artan bir yaklaşımdır.

KAP'ın en basit formülasyonu bile, yüksek kalitede sonuçlar elde etmek için oldukça karmaşık ve çözülmesi zor bir problem türüdür. Bu nedenle, KAP modellerinde yapılan değişiklikler, geliştirilen dinamik ve tekrarlı yaklaşımlar dal-sınır algoritması, kesme düzlemleri gibi klasik eniyileme tekniklerinin kullanımlarına engel olmaya başlamıştır.

KAP tabanlı tesis yerleşim problemleri için önerilen algoritmanın başarısı, öncelikle tesis yerleşim problemleri ile ilgili birçok çalışmada kıyaslama amacıyla kullanılan problemlerle test edilmiş ve TZRE çözümü tüm problemler için bilinen en iyi çözüme ulaşmıştır. Literatürde en sık kullanılan KAP problemleri kütüphanelerinden biri olan Burkard (1998)'ın QAPLIB'indeki problemlerinin içerisinde 77 adet problemde, en iyi çözüme yakınsama oranı toplamı olarak %83 olarak elde edilmiştir. Bu yaklaşımın daha iyi sonuçlar vermesi için probleme özel kontrol parametre seçiminin önemli olduğu anlaşılmıştır. Bununla birlikte, yapılan deney çalışması neticesinde, belli aralıklar dahilinde parametre ayarlaması yapmanın problemlerin çoğunda, eniyi/iyi çözüme yakınsama oranını değiştirmediği ortaya çıkmıştır.

Geleneksel KAP'ın *"herhangi bir bölüm herhangi bir yerleşim bölgesine atanabilir"* varsayımı altında çözümü, pratikte uygulanacak problemler için bölüme özel kısıtları olan KAP ile ilgili bir çalışma yapmaya yönlendirmiştir. QAPLIB problemlerinin bir kısmı, bu kısıtları temsil etmek amacıyla, %10, %20, %30 gibi üç farklı kısıt düzeyinde yeniden çözülmüştür.

Bölüme özel kısıtı olan/olmayan KAP tabanlı tesis yerleşim problemleri için önerilen TZRE algoritması kullanılarak geniş çapta yapılan deneysel çalışmalar sonucunda, TZRE'nin KAP tabanlı yerleşim problemlerine uygulanabileceği ve tesis yerleşim problemlerine etkin çözümler bulmak için etkili bir yaklaşım olduğu ortaya çıkmaktadır.

Bu çalışma, mevcut tesis yerleşimi algoritmalarının problemi uzun dönemli ve statik bir bakış açısıyla değerlendirmelerinin yanısıra, TZRE'nin tesis yerleşim problemlerine uygulandığı ilk çalışmalardan biri olması nedeniyle de ayrı bir öneme sahiptir.

TZRE kontrol parametrelerinin, kullanıldıkları probleme bağlı olarak seçilmesi gerektiği düşünülerek; bu çalışma; geniş kapsamlı ve büyük aralıklar için yapılacak parametre seçimi ve yakınsama analizi ile genişletilebilir.

Yapılan deneysel çalışmalar süresince, bölümler arası akış ve uzaklık matrislerinin seyrekliği ile parametre seçimi ve parametre ayarı arasında muhtemel bir ilişki olabileceği göze çarpmıştır. Bir sonraki adım olarak KAP tabanlı yerleşim problemleriyle ilgili bu etkileşimin araştırılması da yararlı olacaktır.

TZRE'nin sürekli problemlere başarıyla uygulanması ve literatürde kesikli problemlerle ilgili çok fazla çalışmanın bulunmaması; TZRE'de, kesikli problemlere özel operatörlerin geliştirilmesinin, bu tekniğin başarısını büyük ölçüde arttırabileceği fikrini doğurmuştur. Kesikli problemlere özel TZRE operatörlerinin geliştirilmesi yada mevcut operatörlerin kesikli problemler için yeniden gözden geçirilmesi bundan sonra yapılacak çalışmalarda ele alınmalıdır.

KAYNAKLAR DİZİNİ

Balakrishnan, J. ve Cheng, C. H., 2000, Genetic search and the dynamic layout problem, Computers and Operations Research, 27, 587-593.

Balakrishnan, J., Cheng, C.H., Conway, D.G., Lau, C.M., 2003, A hybrid genetic algorithm for the dynamic plant layout problem, International Journal of Production Economics, 86, 107-120.

Baykasoglu, A. ve Gindy, N.N.Z, 2001. A simulated annealing algorithm for dynamic layout problem, Computers and Operations Research, 28, 14, 1403–1426.

Bergh, F. ve Engelbrecht, A.P., 2005, A Study of particle swarm optimization particle trajectories, Automation in Construction, 14, 3, 393-404.

Burkard, R.E., Cela, E. and Karisch, S.E. 1996, QAPLIB (Quadratic Assignment Problems Library): http://www.opt.math.tu-graz.ac.at/qaplib/ (retrieved on September 2005).

Burkard, R.E., Çela, E., Pardalos, P.M. ve Pitsoulis,L., 1998, The quadratic assignment problem, Pardalos, P.M., Du,D.-Z.(Derl.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, 241–338.

Chiang, W.C. ve Chiang, C., 1998, Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation, European Journal of Operational Research, 106, 1, 457-488.

Clerc M., 2000, Discrete Particle Swarm Optimization Illustrated by the Traveling Salesman Problem, http://www.mauriceclerc.net

Dorigo, M., Maniezzo, V., and Colorni, A., 1996 Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26 (1), 29-41.

Dunker,T., Radons, G. and Westkamper, E., 2005, Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem, European Journal of Operational Research, 165, 1, 55-69.

El-Sharkawi, M.A., 2006, Particle Swarm Optimization, http://cialab.ee.washington.edu/index_files/tutorial /TZRE.pdf

Erel, E., Ghosh, J.B. ve Simon J.T., 2003, New heuristic for the dynamic layout problem, Journal of the Operational Research Society ,54, 12, 1275-1282.

Fukuyama, Y., 2006, Fundamentals of Particle Swarm Optimization Techniques , http://homepage2.nifty.com/fukuyama-yoshikazu/ECTTutorial.pdf

Gambardella, L. M., Taillard, E.D. ve Dorigo, M, 1999, Ant colonies for the QAP, Journal of the Operations Research Society, 50, 67–76.

Golany, B. ve Rosenblatt, M. J., 1989, A heuristic algorithm for the quadratic assignment formulation to the plant layout problem, International Journal of Productions Research, 27, 2, 293-308.

Hu S., 2006, TZRE Tutorial, http://www.swarmintelligence.org/tutorials.php

Hu, X., Shi, Y. ve Eberhart, R., 2004, Recent Advances in Particle Swarm, IEEE congress on Evolutionary Computation, Portland, Oregon, USA.

İşlier, A.A., 1997, Tesis Planlaması Ders Notları,192s.

İşlier, A.A., 1998, A genetic algorithm approach for multiple criteria facility layout design, International Journal of Production Research, 36, 1549–1569.

Kapanoglu M. ve Utkan F., 2006, Particle Swarm Optimization for Facility Layout Departments with/out Department Specific Restrictions, Lecture Notes in Computer Science-4150, 516-517.

Kapanoglu M., Utkan F. ve Alikalfa M., 2005, Particle Swarm Optimization for Facility Layout Problems", Proceedings of International Conference on Industrial Engineering Theory, Applications & Practice, Clearwater Beach, Florida, USA

Kapanoglu M. ve Utkan F., Particle Swarm Optimization for Facility Layout Problems, International Conference on Computers and Industrial Engiennering, 19-22 June 2005, Istanbul, Türkiye.

Kennedy, J. ve Eberhart, R. C., 1995, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks, 1942-1948.

Kennedy, J., Eberhart R. C., ve Shi, Y., 2001, Swarm Intelligence, Morgan Kauffman, San Diego.

Kochhar, J.S., Foster B.T., Heragu, S.S., 1998, Hope: A genetic algorithm for the unequal area facility layout problem, Computers and operations Research, 25, 7-8, 583-594.

Koopmans, J. and Beckmann, M. 1967. Assignment problems and location of economic activities. Econometrica, 25, 53–76.

Lacksonen, T.A. ve Enscore, E.E., 1993, Quadratic assignment algorithms for the dynamic layout problem, International Journal of Productions Research, 31, 3, 503-517.

Lacksonen, T.A., 1997, Preprocessing for static and dynamic facility layout problems, International Journal of Production Research 35, 4, 1095–1106.

Liggett, R.S., 2000, Automated Facilities Layout: Past, Present, Future, Automation in Construction, 9, 197–215.

Loila, E.M., Abreu, N.M.M., Netto, P.O.B., Hahn, P. ve Querido, T., 2005, A survey for the quadratic assignment problem, European Journal of Operational Research (baskıda).

McKendall, A.R.Jr. and Shang, J. 2006. Hybrid ant systems for the dynamic facility layout problem. Computers and Operations Research, 33(3), 790-803.

Meller, R. D. and Gau, K-Y., 1996. The facility layout problem: Recent emerging trends and perspective, Journal of Manufacturing Systems, 15, 5, 351.

Norman B. A. and Smith A. E.,1997. Random Keys Genetic Algorithm with Adaptive Penalty Function for Optimization of Constrained Facility Layout Problems, Proceedings of the IEEE Conference on Evolutionary Computing, 407-411.

Penev, K. ve Littlefair, G., 2005, Free search-a comparative analysis, Information Sciences, 172, 1-2, 173-193.

Rosenblatt, M., 1986, The dynamics of plant layout, Management Science, 32, 1, 76-86.

Salman, A., Ahmad, I. ve Al-Madani, S., 2002, Particle swarm optimization for task assignment problem, Microprocessors and Microsystems, 26, 8, 363-371.

Skorin-Kapov, J., 1990, Tabu search applied to the quadratic assignment problem, ORSA Journal on Computing, 2, 1, 33-45.

Solimanpur M., Vrat, P. ve Shankar R., 2005, An ant algorithm for the single row layout problem in flexible manufacturing systems , Computers and Operations Research, 32, 583–598.

Su, B.A. ve Aslan, D., 1997, Tesis Planlama, Dokuz Eylül Üniversitesi Mühendislik Fakültesi Yayınları, 201, 152s.

Tate, D.M. ve Smith, A.E., 1995, Unequal- area facility layout by genetic search, IIE Transactions, 27, 4, 465.

Tompkins J.A. ve White, J.A., 1984, Facilities Planning, John Wiley&Sons Inc.

Tompkins, J.A.., 1997, Facilities Planning: a vision for the 21st century, IIE Solutions, August.

Tompkins J.A., White, J.A., Bozer, Y.A. ve Tanchoco J.M.A, 2003, Facilities Planning, John Wiley&Sons Inc., 750s.

Trelea, I.C., 2003, The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85 317-325.

Üstün, Ö., 2001, Kareli Atama Probleminin Çözümü İçin Subgradient Yöntemi, Yüksek Lisans Tezi, Osmangazi Üniversitesi Mühendislik-Mimarlık Fakültesi, 88s.

Wang, M.J., Hu, M.H. ve Ku, M.Y., 2005, A solution to the unequal area facilities layout problem by genetic algorithm, Computers in Industry, 56, 207–220.

Xia, W. Ve Wu, Z., 2005, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, Computers&Industrial Engineering, 48,409–425.

EKLER

EK.1. Tesis yerleşim problemleri ile ilgili 1986-1996 arasında yayınlanmış makale bilgileri

EK.2. TZRE algoritması için geliştirilen program kodları

EK.3. İlk test problem setinin ayrıntılı sonuçları

EK.4. QAPLIB problem setinin ayrıntılı sonuçları

EK.5. QAPLIB problemleri için TZRE algoritmalarını ve bölüm büyüklüklerini karşılaştırma tablosu

EK.6. Bölüme özel kısıtları olan problemlerden birinin LINGO'da modellenmiş örneği

EK.1

Yazarlar	Yıl	Kategori	Notlar
Aneke ve Carrie	1986	C.1.b	Flowline layout heuristic
Gupta	1986	B.3.a	Simulation of layout flexibility
Hassan, Hogg ve Smith	1986	A.2.a	SHAPE
Rosenblatt	1986	B.1.a	Dynamic layout problem
Rosenblatt ve Sinuany-Stern	1986	B.3.a	Robust layout selection
Drezner	1987	A.2.b	Scatter diagram based on layout construction
Evans, Wilhelm ve Karwowski	1987	A.2.d	Fuzzy logic based layout construction
Giffin ve Foulds	1987	A.2.b	Graph-theoretic, continous adjacency measure
Hassan ve Hogg	1987	A.1.b	Graph-theoeritic review
Jacobs	1987	B.3.a	Multiple-criteria
Montreuil, Ratliff ve Goetschalekx	1987	A.2.b	MATCH
Rosenblatt ve Lee	1987	B.3.b	Robust layout evaluation for QAP
Urban	1987	B.3.b	Weighted-objective model
Wemmerlow ve Hyer	1987	C.3.a	Cellular manufacturing layout
Wilhelm ve Ward	1987	A.2.d	Simulated annealing for QAP
Co ve Araar	1988	C.3.a	Cellular manufacturing layout
Heragu ve Kusiak	1988	C.2.a	Machine layout in FMS
Kaku, ThomTZREn ve Baybars	1988	A.2.d	Multifloor QAP
Smith ve MacLeod	1988	A.1.a	Quadratic set packing problem relaxation
Malakooti	1989	B.3.b	Multiobjective heuristic
Malakooti ve Tsurushima	1989	B.3.a	Expert system to choose layout
Montreuil ve Ratliff	1989	A.1.a	Cut trees for layout
Urban	1989	B.3.a	Multiobjective
Wemmerlow ve Hyer	1989	C.3.a	Cellular manufacturing application survey
Heragu ve Kusiak	1990	C.2.b	Machine layout in FMS
Kouvelis ve Kiran	1990	C.2.a	Modified QAP for manufacturing
Montreuil	1990	A.1.c	MIP Approach
Vakharia ve Wemmerlow	1990	C.3.a	Cellular manufacturing design
Al-Hakim	1991	A.2.b	Graph-theoritic construction
Foulds	1991	A.1.b	Graph-theoritic text
Hassan ve Hogg	1991	A.1.b	Block layout from graph
Heragu ve Kusiak	1991	A.1.c	MIP formulation
Huntley ve Brown	1991	A.2.d	Parallel computation for QAP
Kaku, ThomTZREn ve Morton	1991	A.2.d	QAP heuristic
Kouvelis ve Kiran	1991	B.1.a	Dynamic layout models
Logendran	1991	C.3.a	Operation and machine sequence
Montreuil ve Venkatadri	1991	B.1.a	Dynamic layout requirements
Raoot ve Rakshit	1991	A.2.a	Placement procedures in heuristics
Tam ve Li	1991	A.2.a	HAL
Van Camp, Carter ve Vannelli	1991	A.1.a	Nonlinear optimization
Al-Hakim	1992	A.2.b	Block layout from graph
Balakrishnan, Jacobs ve Venkataramanan	1992	B.1.b	Dynamic facility layout for QAP

EK.1 (Devam)

Yazarlar	Yıl	Kategori	Notlar
Banerjee vd.	1992	A.2.c	Reasoning-based construction
Boswell	1992	A.2.b	Graph-theoritic construction-TESSA
Chbajed, Montreuil ve Lowe	1992	A.1.a	Flow network design
Goetschalckx	1992	A.2.b	SPIRAL
Harmonsky ve Toethero	1992	B.3.b	Multiobjective QAP heuristic
Heragu ve Alfa	1992	C.1.b	Simulated annealing for row layout
Irani, Coben ve Cavalier	1992	C.3.a	Simultaneous cell formation and layout
Jajodia vd	1992	A.2.d	Simulated annealing for QAP
Kaku ve Rachamadugu	1992	C.1.b	Loop line design of FMS
Kouvelis ve Chiang	1992	C.1.b	Simulated annealing for row layout
Kouvelis, Kurawarwala ve Gutierrez	1992	B.3.b	Robust layout models
Leung	1992	C.1.b	Graph-theoritic construction
Montreuil ve Laforge	1992	B.1.a	Dynamic facility layout
Palekar vd.	1992	B.1.a	Stochastic-dynamic QAP models
Rosenblatt ve Golany	1992	A.1.a	Alternative QAP-approach
Rosenblatt ve Kropp	1992	B.2.a	Stochastic layout problem
Sarin vd.	1992	B.3.a	Decision-theoretic approach for QAP
Tam	1992	A.2.a	Simulated annealing and genetic algorithm heuristics
Urban	1992	B.1.a	Bounds for dynamic layout
Das	1993	C.3.b	Flexible manufacturing system layout
Ho, Lee ve Moodie	1993	C1.b	Multi-flow line layout
Irani, Cavalier ve Cohen	1993	C.3.a	Layout and virtual manufacturing cells
Liao	1993	C.1.b	Cellular manufacturing along line
Montreuil, Venkatadri ve Ratliff	1993	A.2.c	Design skeleton for MIP
Shang	1993	B.3.b	Hierarcical multicriterion
Suresh ve Sahu	1993	B.3.b	Simulated annealing, multiobjective QAP
Tate ve Smith	1993	A.2.a	FLEX-BAY
Urban	1993	B.1.b	Dynamic facility layout heuristic
Welgama ve Gibson	1993	C.2.b	Machine layout problem
Bozer, Meller ve Erlebacher	1994	A.2.a	MULTIPLE
Conway ve Venkataramanan	1994	B.1.b	Genetic algorithm for dynamic QAP
Heragu ve Gupta	1994	C.3.b	Cellular manufacturing design
Lacksonnen	1994	A.2.c	Dynamic layout MIP
Langevin, Montreuil ve Riopel	1994	C.1.a	Spine-based layout
Sirinaovakul ve Thajcbayapong	1994	A.2.d	Knowledge-based, fixed-shape departments
Skorin-Kapov	1994	A.2.d	Tabu-search for QAP
Tretbeway ve Foote	1994	A.2.b	Scatter diagram to facility layout
Welgama, Gibson ve Al-Hakim	1994	A.2.b	Knowledge-based graph layout
Banerjee ve Zhou	1995	C.1.a	Single-loop layout
Hassan	1995	A.2.b	Group technology layouts
Kim ve Kim	1995	A.2.b	Graph-theoritic, distance based
Kouvelis, Chiang ve Yu	1995	C.1.a	Row layout models
Sarker vd.	1995	C.1.a	One-dimensional machine location
Souilah	1995	A.2.d	Hierarchical simulated annealing design
Benjaafar, Sheikhzadeh ve Soewito	1996	B.3.a	Flexible layouts
Chiang ve Kouvelis	1996	A.2.d	Tabu-search

EK.1 (Devam)

Yazarlar	Yıl	Kategori	Notlar
Fu ve Kaku	1996	A.1.a	Work-in-process vs. layout
Liao vd.	1996	C.3.a	Cellular manufacturing along line
Meller ve Bozer	1996	A.2.a	Simulated annealing algorithm
Meller ve Gau	1996	B.3.a	Layout objectives study
Goldschmidt, Takvorian ve Yu	1996	A.1.b	Graphical representation

EK.2

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

public struct Params
{
      public double W;
      public double c1;
      public double c2;
      public int Vmin;
      public int Vmax;
}

            /// <summary>
            /// The main entry point for the application.
            /// </summary>
            [STAThread]
            static void Main()
            {
                  Application.Run(new Form1());
            }
            double [,] TZRE;
            double [,] TZREtemp;
            int [,] Dept;
            int [,] PBest;
            double [] PBestFitness;
            double [,] Velocity;
            double [,] VelocityNew;
            double [] Fitness;
            int [] GBest;
            double GBestFitness;
            double TZREAverage;
            Params TZREParams;
            Random rd;

            private void button1_Click(object sender, System.EventArgs e)
            {
                  Init();

                  int iteration;
                  iteration=Convert.ToInt32(textBox9.Text);

                  for (int i = 1; i <= iteration ; i++)
                  {
                        TZREtoDept();
                        Evaluate();
                        CalculateVelocity();
                        PrepareTZRE();
```

EK.2 (Devam)

```
                        textBox10.Text=i.ToString();
                        //Chart Options

      Chart.ChartGroups.Group0.ChartData.SeriesList[0].PointData[i] = new
PointF(i,(float) GBestFitness);

      Chart.ChartGroups.Group0.ChartData.SeriesList[1].PointData[i] = new
PointF(i,(float) TZREAverage);
                        Application.DoEvents();
                  }

                  string a="";
                  for (int i = 0; i <= GBest.GetUpperBound(0); i++)
                  {
                        a=a+"-"+GBest[i].ToString();
                  }
                  textBox13.Text=a;
                  textBox14.Text=GBestFitness.ToString();

            }

            public void Init()
            {
                  TZRE = new double
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  TZREtemp = new double
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  Velocity = new double
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  VelocityNew = new double
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  Dept = new int
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  PBest = new int
[Convert.ToInt32(textBox6.Text),Convert.ToInt32(textBox7.Text)];
                  Fitness = new double [Convert.ToInt32(textBox6.Text)];
                  PBestFitness = new double
[Convert.ToInt32(textBox6.Text)];
                  GBest = new int [Convert.ToInt32(textBox7.Text)];
                  rd=new Random(Convert.ToInt32(textBox8.Text));

                  GBestFitness=999999999;

                  for (int i = 0; i <= PBestFitness.GetUpperBound(0) ;
i++)
                  {
                        PBestFitness[i]=99999999999;
                  }

                  TZREParams.Vmin = Convert.ToInt32(textBox4.Text);
                  TZREParams.Vmax = Convert.ToInt32(textBox5.Text);
```

EK.2 (Devam)

```
                  TZREParams.W = Convert.ToDouble(textBox1.Text);
                  TZREParams.c1 = Convert.ToDouble(textBox2.Text);
                  TZREParams.c2 = Convert.ToDouble(textBox3.Text);

                  for (int i = 0; i <= TZRE.GetUpperBound(0); i++)
                  {
                        for (int j = 0; j <=TZRE.GetUpperBound(1) ; j++)
                        {
                              TZRE[i,j]=rd.NextDouble();
                              Dept[i,j]=0;
                        }
                  }

                  for (int i = 0; i <=Velocity.GetUpperBound(0) ; i++)
                  {
                        for (int j = 0; j <= Velocity.GetUpperBound(1) ;
j++)
                        {

      Velocity[i,j]=rd.Next(TZREParams.Vmin,TZREParams.Vmax+1);
                              VelocityNew[i,j]=0;
                        }
                  }

                  //Chart Initialization

      Chart.ChartGroups.Group0.ChartData.SeriesList[0].PointData.Clear();

      Chart.ChartGroups.Group0.ChartData.SeriesList[1].PointData.Clear();

      Chart.ChartGroups.Group0.ChartData.SeriesList[0].PointData.Length=C
onvert.ToInt32(textBox9.Text)+1;

      Chart.ChartGroups.Group0.ChartData.SeriesList[1].PointData.Length=C
onvert.ToInt32(textBox9.Text)+1;


            }

            public void TZREtoDept()
            {
                  //Bütün sayýlar iþleniyor.
                  double sayi;
                  for (int i = 0; i <= TZRE.GetUpperBound(0) ; i++)
                  {
                        sayi=TZRE[i,0];
                        for (int j = 1; j <= TZRE.GetUpperBound(1) ; j++)
                        {
                              if (sayi<TZRE[i,j])
                              {
                                    sayi=TZRE[i,j];
```

EK.2 (Devam)

```
                              }
                        }
                        for (int j = 0; j <= TZRE.GetUpperBound(1) ; j++)
                        {
                              TZRE[i,j]=TZRE[i,j]-sayi;
                        }
                  }

                  for (int i = 0; i <= TZRE.GetUpperBound(0) ; i++)
                  {
                        for (int j = 0; j <= TZRE.GetUpperBound(1) ; j++)
                        {
                              TZREtemp[i,j]=TZRE[i,j];
                        }
                  }

                  for (int i = 0; i <= TZRE.GetUpperBound(0) ; i++) //
Bütün particle lar için
                  {
                        for (int j = 0; j <= TZRE.GetUpperBound(1) ; j++)
// Dept kere
                        {
                              double deger=TZREtemp[i,0];
                              int indis=0;
                              for (int k = 0; k <= TZRE.GetUpperBound(1) ;
k++) //Bütün deðerlere bak.
                              {
                                    if (TZREtemp[i,k]<deger)
                                    {
                                          indis=k;
                                          deger=TZREtemp[i,k];
                                    }
                              }
                              TZREtemp[i,indis]=99999999;
                              Dept[i,indis]=j+1;
                        }
                  }

            }

            public void Evaluate()
            {
                  //Layout un fitnessi hesaplanacak.
                  //EVALUATE //Eðer Tek Sýra Layout varsa

                        for (int i = 0; i <= Dept.GetUpperBound(0) ; i++)
// Her particle için
                        {
                              double uyum=0;
                              for (int j = 0; j <
Convert.ToInt32(textBox7.Text) ; j++)
```

EK.2 (Devam)

```
                                {
                                      for (int k = 0; k <
Convert.ToInt32(textBox7.Text) ; k++)
                                      {
                                            int k1=Dept[i,j];
                                            int k2=Dept[i,k];
                                            int
a=Convert.ToInt32(Flow.Sheets[0].Cells[k1-1,k2-1].Value);
                                            //int b=Math.Abs(j-k);
                                            double
b=Convert.ToDouble(Dist.Sheets[0].Cells[j,k].Value);
                                            uyum=uyum+a*b;
                                      }
                                }
                                Fitness[i]=uyum;
                          }

                    //Find PBests
                    for (int i = 0; i <= PBestFitness.GetUpperBound(0) ;
i++)
                    {
                          if (PBestFitness[i]<Fitness[i])
                          {
                                PBestFitness[i]=Fitness[i];
                                for (int j = 0; j <= Dept.GetUpperBound(1) ;
j++)
                                {
                                      PBest[i,j]=Dept[i,j];
                                }
                          }

                    }

                    //FIND Global BEST
                    double deger=Fitness[0];
                    int indis=0;
                    for (int i = 1; i <= Fitness.GetUpperBound(0) ; i++)
                    {
                          if (Fitness[i]<deger)
                          {
                                indis=i;
                                deger=Fitness[i];
                          }
                    }
                    if (deger<GBestFitness)
                    for (int i = 0; i <= Dept.GetUpperBound(1) ; i++)
                    {
                          GBest[i]=Dept[indis,i];
                          GBestFitness=deger;
                    }
```

EK.2 (Devam)

```
                  //Find Average
                  double toplam=0;
                  for (int i = 0; i <= Fitness.GetUpperBound(0) ; i++)
                  {
                        toplam=toplam+Fitness[i];
                  }
                  TZREAverage=toplam/Convert.ToInt32(textBox6.Text);

            }

            public void CalculateVelocity()
            {
                  for (int i = 0; i <= Velocity.GetUpperBound(0) ; i++)
                  {
                        for (int j = 0; j <= Velocity.GetUpperBound(1) ;
j++)
                        {
                              double rnd1=rd.NextDouble();
                              double rnd2=rd.NextDouble();
                              double
deger=TZREParams.W*Velocity[i,j]+TZREParams.c1*rnd1*(PBest[i,j]-
Dept[i,j])+TZREParams.c2*rnd2*(GBest[j]-Dept[i,j]);
                              double velo =
deger/(TZREParams.W+TZREParams.c1*rnd1+TZREParams.c2*rnd2);
                              if (checkBox1.Checked==true)
                              {
                                    //Kýsýtla
                                    if (TZREParams.Vmin<=velo &
velo<=TZREParams.Vmax)
                                    {
                                          VelocityNew[i,j]=velo;
                                    }
                                    if (velo<TZREParams.Vmin)
                                    {

      VelocityNew[i,j]=TZREParams.Vmin;
                                    }
                                    if (velo>TZREParams.Vmax)
                                    {

      VelocityNew[i,j]=TZREParams.Vmax;
                                    }
                              }
                              else
                              {
                                    //Kýsýtlama
                                    VelocityNew[i,j]=velo;
                              }

                        }
                  }
```

EK.2 (Devam)

```
                  //Yeni velocity eskisinin yerine konacak.
                  for (int i = 0; i <= Velocity.GetUpperBound(0) ; i++)

                  {
                        for (int j = 0; j <= Velocity.GetUpperBound(1) ;
j++)
                        {
                              Velocity[i,j]=VelocityNew[i,j];
                        }
                  }
            }

            public void PrepareTZRE()
            {
                  //Yeni Velocity e göre TZRE'yu hesapla.
                  //Real value lar toplandý. (Department number lar
deðil)
                  for (int i = 0; i <= TZRE.GetUpperBound(0) ; i++)
                  {
                        for (int j = 0; j <= TZRE.GetUpperBound(1) ; j++)
                        {
                              TZRE[i,j]=Velocity[i,j]+TZRE[i,j];
                        }
                  }

            }

            private void button2_Click(object sender, System.EventArgs e)
            {
                  Flow.OpenExcel(textBox11.Text);
            }

            private void Form1_Load(object sender, System.EventArgs e)
            {

            }

            private void button3_Click(object sender, System.EventArgs e)
            {
                  Dist.OpenExcel(textBox15.Text);

            }

            private void button4_Click(object sender, System.EventArgs e)
            {

      Chart.SaveImage(textBox12.Text,System.Drawing.Imaging.ImageFormat.B
mp);
            }

      }
```

EK.3

1. Rosenblatt (1986)

$$Flow = \begin{Bmatrix} 0 & 63 & 605 & 551 & 116 & 136 \\ 63 & 0 & 635 & 941 & 50 & 191 \\ 104 & 71 & 0 & 569 & 136 & 55 \\ 65 & 193 & 622 & 0 & 77 & 90 \\ 162 & 174 & 607 & 591 & 0 & 179 \\ 156 & 13 & 667 & 611 & 175 & 0 \end{Bmatrix}$$

Optimal Solution

1	3	5
6	4	2

Cost = 12,822

TZRE Solution

1	3	5
6	4	2

Cost = 12,822

2. Rosenblatt (1986)

$$Flow = \begin{Bmatrix} 0 & 175 & 804 & 904 & 56 & 176 \\ 63 & 0 & 743 & 936 & 45 & 177 \\ 168 & 85 & 0 & 918 & 138 & 134 \\ 51 & 94 & 962 & 0 & 173 & 39 \\ 97 & 104 & 730 & 634 & 0 & 144 \\ 95 & 115 & 983 & 597 & 24 & 0 \end{Bmatrix}$$

Optimal Solution

1	4	2
5	3	6

Cost = 14,853

TZRE Solution

1	4	2
5	3	6

Cost = 14,853

3. Rosenblatt (1986)

$$Flow = \begin{Bmatrix} 0 & 90 & 77 & 553 & 769 & 139 \\ 168 & 0 & 114 & 653 & 525 & 185 \\ 32 & 35 & 0 & 664 & 898 & 87 \\ 27 & 166 & 42 & 0 & 960 & 179 \\ 185 & 56 & 44 & 926 & 0 & 104 \\ 72 & 128 & 173 & 634 & 687 & 0 \end{Bmatrix}$$

Optimal Solution

1	5	3
2	4	6

Cost = 13,172

TZRE Solution

1	5	3
2	4	6

Cost = 13,172

4. Rosenblatt (1986)

$$Flow = \begin{Bmatrix} 0 & 112 & 15 & 199 & 665 & 649 \\ 153 & 0 & 116 & 173 & 912 & 671 \\ 10 & 28 & 0 & 182 & 855 & 542 \\ 29 & 69 & 15 & 0 & 552 & 751 \\ 198 & 71 & 42 & 24 & 0 & 758 \\ 2 & 109 & 170 & 90 & 973 & 0 \end{Bmatrix}$$

Optimal Solution

1	6	4
2	5	3

Cost = 13,032

TZRE Solution

1	6	4
2	5	3

Cost = 13,032

EK.3 (Devam)

5. Rosenblatt (1986)

$$Flow = \begin{Bmatrix} 0 & 663 & 23 & 128 & 119 & 50 \\ 820 & 0 & 5 & 98 & 141 & 66 \\ 822 & 650 & 0 & 137 & 78 & 91 \\ 826 & 570 & 149 & 0 & 93 & 151 \\ 915 & 515 & 53 & 35 & 0 & 177 \\ 614 & 729 & 178 & 10 & 99 & 0 \end{Bmatrix}$$

Optimal Solution

3	2	6
4	1	5

Cost = 12,819

TZRE Solution

3	2	6
4	1	5

Cost = 12,819

6. Zimmermann and Sovereign (1974)

$$Flow = \begin{Bmatrix} 0 & 2 & 7 & 1 \\ 2 & 0 & 4 & 2 \\ 6 & 4 & 0 & 3 \\ 1 & 2 & 3 & 0 \end{Bmatrix} \quad Distance = \begin{Bmatrix} 0 & 10 & 36 & 9 \\ 19 & 0 & 4 & 4 \\ 5 & 2 & 0 & 8 \\ 70 & 26 & 1 & 0 \end{Bmatrix}$$

Optimal Solution: (4,2,3,1) Cost: 389

TZRE Solution: (4,2,3,1) Cost:389

7. Gavett (1968)

$$Flow = \begin{Bmatrix} 0 & 6 & 7 & 2 \\ 6 & 0 & 5 & 6 \\ 7 & 5 & 0 & 1 \\ 2 & 6 & 1 & 0 \end{Bmatrix} \quad Distance = \begin{Bmatrix} 0 & 10 & 20 & 5 \\ 18 & 0 & 9 & 4 \\ 5 & 5 & 0 & 8 \\ 8 & 0 & 15 & 0 \end{Bmatrix}$$

Optimal Solution (2,4,3,1) Cost: 403

TZRE Solution: (2,4,3,1) Cost: 403

EK.4

No	Problem	Çözüm Yöntemi	Uyum Değeri	Yerleşim	Opt. (%)
1	Bur26a	PSO-St	5549427	-7-20-13-11-19-5-15-12-17-24-25-22-4-21-6-18-26-3-8-14-9-23-10-16-1-2	98
		PSO-VR	5577499	-26-15-18-7-19-24-12-1-14-3-16-23-22-17-9-25-21-13-11-10-4-2-8-5-20-6	97
		PSO-M	5543157	-9-6-1-23-13-24-7-16-19-21-15-14-2-17-11-26-22-12-18-20-10-8-3-4-5-25	98
		En iyi çözüm	5426670	-26 15 11 7 4 12 13 2 6 18 1 5 9 21 8 14 3 20 19 25 17 10 16 24 23 22	
2	Bur26b	PSO-St	3925079	-7-17-8-13-18-26-20-2-11-5-12-22-1-19-25-6-24-23-3-16-14-4-15-21-10-9	97
		PSO-VR	3901073	-7-24-15-13-19-25-2-17-14-8-23-3-5-12-10-21-6-22-18-11-9-20-1-4-16-26	98
		PSO-M	3908419	-11-21-1-15-14-26-2-13-17-10-20-9-8-19-4-24-5-6-18-12-25-16-7-3-22-23	98
		Grasp	3817852		
3	Bur26c	PSO-St	5574281	-26-16-15-13-19-22-25-20-10-6-9-12-3-21-18-23-4-1-5-17-7-24-2-8-14-11	97
		PSO-VR	5615247	-25-7-5-22-19-12-26-13-16-6-18-23-11-21-17-15-1-4-20-14-8-10-3-9-2-24	97
		PSO-M	5583668	-16-7-20-15-19-2-17-26-8-25-6-22-21-23-18-5-3-11-1-13-14-12-10-4-24-9	97
		Grasp	5426795		
4	Bur26d	PSO-St	3925173	-6-17-21-14-10-3-25-13-8-1-9-2-15-5-26-23-16-22-18-11-19-4-12-7-24-20	97
		PSO-VR	3939160	-17-23-5-2-19-13-15-8-7-6-26-21-1-18-9-10-22-14-24-12-3-16-25-4-20-11	97
		PSO-M	3938374	-8-4-20-12-18-26-11-14-19-24-9-23-1-15-10-5-7-3-2-13-21-6-22-25-17-16	97
		Grasp	3821225		
5	Bur26e	PSO-St	5535373	-18-6-26-3-17-7-24-22-14-16-20-21-2-12-13-10-23-15-19-11-9-1-5-4-8-25	97
		PSO-VR	5524089	-10-7-12-1-19-22-8-21-17-15-3-4-20-18-6-11-16-13-24-14-23-2-25-9-5-26	98
		PSO-M	5568921	-25-16-24-13-7-3-22-21-18-8-11-4-15-5-17-9-10-14-19-1-12-6-26-2-20-23	97
		Grasp	5386879		
6	Bur26f	PSO-St	3882849	-10-2-23-3-17-16-20-6-18-13-26-21-15-5-7-25-8-4-14-12-19-22-1-9-24-11	97
		PSO-VR	3915030	-16-10-22-15-9-24-6-3-19-1-8-12-21-18-25-20-7-11-23-13-4-2-17-5-26-14	97
		PSO-M	3901009	-9-26-14-2-17-15-19-23-10-25-8-3-5-4-18-24-22-11-20-12-16-1-13-21-7-6	97
		Grasp	3782044		
7	Bur26g	PSO-St	10363373	-8-2-22-11-18-24-20-26-6-3-7-23-12-14-17-9-16-4-19-21-13-1-25-10-15-5	98
		PSO-VR	10406493	-10-5-8-20-19-17-24-2-6-3-15-12-11-18-9-14-26-23-22-13-16-1-21-4-25-7	97
		PSO-M	10323363	-7-4-24-1-17-5-14-12-19-6-26-11-22-21-8-10-15-3-18-13-16-23-20-25-2-9	98
		Grasp	10117172		

EK.4 (Devam)

8	Bur26h	PSO-St	7281467	-8-2-22-11-18-24-20-26-6-3-7-23-12-14-17-9-16-4-19-21-13-1-25-10-15-5	97
		PSO-VR	7311706	-18-11-24-6-19-26-12-23-13-3-25-22-21-20-10-7-8-5-14-17-16-2-4-9-15-1	97
		PSO-M	7281467	-8-2-22-11-18-24-20-26-6-3-7-23-12-14-17-9-16-4-19-21-13-1-25-10-15-5	97
		Grasp	7098658		
9	Chr12a	PSO-St	16754	-10-3-12-9-2-11-6-5-7-8-1-4	57
		PSO-VR	14394	-8-3-9-5-11-1-12-6-4-10-2-7	66
		PSO-M	14790	-9-4-12-7-10-1-8-6-3-11-2-5	65
		En iyi çözüm	9552	7,5,12,2,1,3,9,11,10,6,8,4	
10	Chr12b	PSO-St	16996	-7-11-2-12-1-5-6-3-8-4-10-9	57
		PSO-VR	16996	-7-11-2-12-1-5-6-3-8-4-10-9	57
		PSO-M	12432	-10-9-6-7-1-3-5-4-12-11-2-8	78
		En iyi çözüm	9742	(5,7,1,10,11,3,4,2,9,6,12,8)	
11	Chr12c	PSO-St	14246	-6-11-1-9-2-10-4-5-7-3-8-12	78
		PSO-VR	16366	-12-11-7-5-6-8-4-1-3-2-10-9	68
		PSO-M	16948	-5-9-6-7-4-8-1-3-11-2-10-12	66
		En iyi çözüm	11156	(7,5,1,3,10,4,8,6,9,11,2,12)	
12	Chr15a	PSO-St	23138	-15-4-14-9-11-10-1-2-13-6-12-3-8-5-7	43
		PSO-VR	16242	-2-4-9-15-6-10-5-3-11-7-14-12-8-1-13	61
		PSO-M	18152	-4-3-9-8-14-6-1-5-15-10-13-12-7-2-11	55
		En iyi çözüm	9896	(5,10,8,13,12,11,14,2,4,6,7,15,3,1,9)	
13	Chr15b	PSO-St	19220	-7-10-5-11-3-12-15-1-2-14-6-9-8-4-13	42
		PSO-VR	23922	-12-1-7-5-11-14-8-10-15-13-6-9-3-4-2	33
		PSO-M	20162	-5-7-9-12-1-13-15-4-11-3-6-8-10-2-14	40
		En iyi çözüm	7990	(4,13,15,1,9,2,5,12,6,14,7,3,10,11,8)	
14	Chr15c	PSO-St	18596	-15-3-2-11-12-7-5-10-8-6-14-4-1-13-9	51
		PSO-VR	20752	-3-2-9-13-10-4-1-12-5-11-14-8-7-6-15	46
		PSO-M	18426	-11-15-6-9-14-7-13-10-4-1-2-3-8-12-5	52
		En iyi çözüm	9504	(13,2,5,7,8,1,14,6,4,3,15,9,12,11,10)	
15	Chr18a	PSO-St	35202	-13-16-5-18-8-6-10-1-4-12-15-14-3-11-9-2-17-7	32
		PSO-VR	33884	-11-9-13-5-2-7-8-1-10-3-18-17-16-6-14-12-15-4	33
		PSO-M	32346	-16-7-8-11-13-3-12-1-17-2-4-18-9-15-14-6-10-5	34
		En iyi çözüm	11098	(3,13,6,4,18,12,10,5,1,11,8,7,17,14,9,16,15,2)	
16	Chr18b	PSO-St	2398	-11-15-17-9-13-4-8-14-6-16-18-1-7-12-10-5-2-3	64
		PSO-VR	2046	-6-4-8-3-5-2-17-7-1-11-9-10-15-16-12-13-14-18	75
		PSO-M	2308	-4-7-2-6-8-1-3-5-11-12-10-9-18-15-13-14-16-17	66
		En iyi çözüm	1534	(1,2,4,3,5,6,8,9,7,12,10,11,13,14,16,15,17,18)	

EK. 4 (Devam)

17	Chr20a	PSO-St	4836	-6-9-10-4-8-7-18-19-12-15-20-14-16-2-5-17-13-11-1-3	45
		PSO-VR	5546	-20-5-13-11-1-17-4-16-18-3-2-9-19-10-7-15-12-6-8-14	40
		PSO-M	4698	-16-9-19-14-6-7-15-2-10-1-3-8-13-11-5-20-12-18-17-4	47
		En iyi çözüm	2192	(3,20,7,18,9,12,19,4,10,11,1,6,15,8,2,5,14,16,13,17)	
18	Chr20b	PSO-St	5194	-5-6-15-18-11-20-13-19-10-2-12-1-14-16-4-17-9-7-3-8	44
		PSO-VR	5066	-17-6-1-13-7-18-2-14-19-16-10-5-15-20-12-4-9-11-3-8	45
		PSO-M	4978	-10-20-9-15-13-6-11-16-19-14-5-4-8-2-3-18-7-12-1-17	46
		En iyi çözüm	2298	(20,3,9,7,1,12,16,6,8,14,10,4,5,13,17,2,18,11,19,15)	
19	Chr20c	PSO-St	45014	-8-20-12-5-2-9-11-17-3-10-13-6-1-19-7-15-18-16-4-14	31
		PSO-VR	44154	-12-3-11-10-7-13-5-14-2-6-1-4-18-19-16-20-9-8-15-17	32
		PSO-M	49096	-15-10-4-5-2-11-9-20-3-1-6-7-19-17-16-18-13-8-12-14	29
		En iyi çözüm	14142	(12,6,9,2,10,11,3,4,15,18,7,13,16,5,14,17,19,1,8,20)	
20	Chr22a	PSO-St	8852	-22-2-14-5-11-15-3-12-17-19-9-8-10-13-6-4-7-18-21-20-1-16	70
		PSO-VR	8754	18-1-7-14-19-4-12-16-21-8-10-20-3-17-13-6-5-11-15-2-9-22	70
		PSO-M	8414	-10-1-15-4-13-16-5-14-6-21-2-8-22-9-7-18-11-12-17-19-3-20	73
		En iyi çözüm	6156	(15,2,21,8,16,1,7,18,14,13,5,17,6,11,3,4,20,19,9,22,10,12)	
21	Chr22b	PSO-St	8336	-11-15-8-12-18-21-6-1-2-16-4-19-7-10-13-9-17-20-22-5-14-3	74
		PSO-VR	8578	-5-18-4-15-8-14-22-13-16-3-10-11-2-21-9-1-6-12-19-7-20-17	72
		PSO-M	8790	-20-17-8-1-9-13-5-14-18-19-11-22-12-6-10-16-3-21-7-2-4-15	70
		En iyi çözüm	6194	(10,19,3,1,20,2,6,4,7,8,17,12,11,15,21,13,9,5,22,14,18,16)	
22	Chr25a	PSO-St	10938	22-8-15-10-20-18-13-14-23-19-6-7-11-5-4-24-17-3-1-16-25-9-12-2-21	35
		PSO-VR	11536	-15-25-14-19-4-8-11-20-10-18-6-17-3-7-24-9-23-22-1-12-5-13-2-21-16	33
		PSO-M	10938	-22-8-15-10-20-18-13-14-23-19-6-7-11-5-4-24-17-3-1-16-25-9-12-2-21	35
		En iyi çözüm	3796	25,12,5,3,18,4,16,8,20,10,14,6,15,23,24,19,13,1,21,11,17,2,22,7,9	
23	Els19	PSO-St	23866358	11-7-3-13-1-16-10-17-5-19-2-8-14-9-15-12-18-4-6	72
		PSO-VR	23350216	-18-19-7-4-9-16-11-10-5-17-13-3-14-15-2-1-12-8-6	74
		PSO-M	22566128	-18-19-11-10-17-4-5-12-6-9-13-16-14-15-2-1-7-8-3	76
		En iyi çözüm	17212548	9,10,7,18,14,19,13,17,6,11,4,5,12,8,15,16,1,2,3	
24	Esc16a	PSO-St	74	-3-4-12-11-2-10-16-6-13-9-14-7-8-1-15-5	92
		PSO-VR	74	-14-12-4-3-9-8-1-2-15-11-16-13-6-7-10-5	92
		PSO-M	72	-3-12-2-11-8-4-10-1-16-14-13-15-9-6-5-7	94
		En iyi çözüm	68	2,14,10,16,5,3,7,8,4,6,12,11,15,13,9,1	
25	Esc16b	PSO-St	292	-14-13-1-12-15-5-16-10-7-3-8-2-4-6-9-11	100
		PSO-VR	292	-13-5-12-1-10-3-11-2-15-7-16-14-8-9-6-4	100
		PSO-M	292	-14-9-8-11-15-3-1-7-16-4-10-6-13-5-12-2	100
		En iyi çözüm	292	6,3,7,5,13,1,15,2,4,11,9,14,10,12,8,16	

EK.4 (Devam)

26	Esc16c	PSO-St	164	-3-1-5-16-7-15-8-10-2-14-4-6-9-12-11-13	98
		PSO-VR	170	-7-5-8-3-15-16-11-1-9-2-10-4-6-14-12-13	94
		PSO-M	174	-6-8-7-3-10-13-9-11-15-1-16-5-14-12-4-2	92
		En iyi çözüm	160	11,14,10,16,12,8,9,3,13,6,5,7,15,2,1,4	
27	Esc16d	PSO-St	22	-6-1-15-3-7-4-9-2-16-5-8-10-13-14-12-11	73
		PSO-VR	22	-14-15-4-7-3-13-5-16-11-9-1-2-10-12-8-6	73
		PSO-M	20	-16-10-5-11-13-8-14-6-4-7-1-9-3-15-12-2	80
		En iyi çözüm	16	14,2,12,5,6,16,8,10,3,9,13,7,11,15,4,1	
28	Esc16e	PSO-St	32	-1-7-9-6-11-4-10-8-12-13-15-5-3-2-16-14	88
		PSO-VR	32	-9-1-8-7-4-11-6-13-3-14-5-16-2-10-12-15	88
		PSO-M	32	-9-1-8-7-4-11-6-13-3-14-5-16-2-10-12-15	88
		En iyi çözüm	28	16,7,8,15,9,12,14,10,11,2,6,5,13,4,3,1	
29	Esc16f	PSO-St	0	-5-3-9-14-12-8-7-15-2-10-1-4-6-16-13-11	100
		PSO-VR	0	-5-3-9-14-12-8-7-15-2-10-1-4-6-16-13-11	100
		PSO-M	0	-5-3-9-14-12-8-7-15-2-10-1-4-6-16-13-11	100
		En iyi çözüm	0	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	
30	Esc16g	PSO-St	30	-15-9-13-10-11-7-4-3-1-14-12-16-6-5-8-2	87
		PSO-VR	32	-5-13-9-14-2-10-3-15-6-16-7-11-8-1-4-12	81
		PSO-M	28	-12-15-6-7-11-10-5-1-9-14-8-4-16-13-2-3	93
		En iyi çözüm	26	8,11,9,12,15,16,14,10,7,6,2,5,13,4,3,1	
31	Esc16h	PSO-St	1006	-6-5-9-10-15-2-14-13-8-11-7-4-1-16-12-3	99
		PSO-VR	1004	-14-7-15-4-2-5-16-9-13-6-12-11-3-10-1-8	99
		PSO-M	996	-12-15-10-7-14-13-5-9-3-1-8-11-2-16-4-6	100
		En iyi çözüm	996	13,9,10,15,3,11,4,16,12,7,8,5,6,2,1,14	
32	Esc16i	PSO-St	14	-10-15-4-16-11-12-5-6-2-14-3-9-1-13-7-8	100
		PSO-VR	16	-7-15-6-12-8-13-9-14-5-3-4-2-11-1-16-10	88
		PSO-M	14	-1-7-2-6-15-9-12-8-3-13-4-5-16-11-10-14	100
		En iyi çözüm	14	13,9,11,3,7,5,6,2,1,15,4,14,12,10,8,16	
33	Esc16j	PSO-St	10	-14-16-8-3-11-12-5-7-15-2-1-10-4-9-13-6	80
		PSO-VR	10	-14-16-8-3-11-12-5-7-15-2-1-10-4-9-13-6	80
		PSO-M	10	-14-2-1-3-10-16-12-4-9-11-8-15-7-6-5-13	80
		En iyi çözüm	8	8,3,16,14,2,12,10,6,9,5,13,11,4,7,15,1	
34	Had12	PSO-St	1692	-8-6-1-11-4-7-10-2-12-3-9-5	98
		PSO-VR	1686	-10-9-1-12-6-7-8-2-11-3-4-5	98
		PSO-M	1682	-9-8-12-1-7-5-3-4-2-11-6-10	98
		En iyi çözüm	1652	3,10,11,2,12,5,6,7,8,1,4,9	

EK.4 (Devam)

35	Had14	PSO-St	2780	-13-3-2-14-10-11-7-1-12-4-6-5-9-8	98
		PSO(VR)	2778	-13-6-1-14-3-11-10-9-12-5-4-7-2-8	98
		PSO-M	2786	-12-7-9-14-8-10-11-5-13-2-4-3-1-6	98
		En iyi çözüm	2724	8,13,10,5,12,11,2,14,3,6,7,1,9,4	
36	Had16	PSO-St	3816	-3-15-14-2-6-11-7-5-9-13-8-10-16-12-4-1	97
		PSO(VR)	3822	-1-12-16-9-15-7-5-3-2-13-11-8-14-10-4-6	97
		PSO-M	3854	-3-11-16-2-8-5-6-14-4-12-9-7-15-10-13-1	97
		En iyi çözüm	3720	9,4,16,1,7,8,6,14,15,11,12,10,5,3,2,13	
37	Had18	PSO-St	5572	-3-11-8-17-7-10-6-2-1-13-16-15-14-12-4-9-18-5	96
		PSO(VR)	5542	-11-3-1-16-10-9-6-18-14-4-12-7-2-17-5-13-15-8	97
		PSO-M	5472	-8-9-1-14-2-3-10-17-13-7-4-6-18-5-16-15-12-11	98
		En iyi çözüm	5358	8,15,16,6,7,18,14,11,1,10,12,5,3,13,2,17,9,4	
38	Had20	PSO-St	7182	-13-3-7-20-4-12-19-2-16-6-1-10-9-15-14-8-18-5-11-17	96
		PSO(VR)	7172	-12-4-1-14-19-7-5-17-20-3-2-11-18-9-13-15-10-16-8-6	97
		PSO-M	7150	-17-11-14-2-12-19-8-4-18-15-10-6-16-5-1-7-20-3-9-13	97
		En iyi çözüm	6922	8,15,16,14,19,6,7,17,1,12,10,11,5,20,2,3,4,9,18,13	
39	Kra30a	PSO-St	114990	-14-9-21-1-16-11-23-13-24-8-17-28-25-26-2-29-5-4-18-22-27-10-15-20-7-6-12-30-19-3	77
		PSO(VR)	113250	-22-25-21-29-4-24-5-11-10-1-30-26-12-18-19-8-20-15-3-9-6-28-14-7-27-23-17-13-2-16	78
		PSO-M	111780	-24-17-21-5-7-16-15-28-30-1-22-23-25-2-19-29-26-14-27-6-3-18-10-4-13-9-12-20-8-11	80
		En iyi çözüm	88900	26,24,23,16,20,19,6,10,11,2,22,18,7,30,15, 21,25,29,12,9,5,17,1,8,13,28,14,3,4,27	
40	Kra30b	PSO-St	118420	-5-17-16-21-24-27-8-25-9-14-22-7-13-1-2-12-26-11-30-23-19-20-10-28-18-3-4-15-29-6	77
		PSO(VR)	115940	-12-8-7-14-13-4-30-15-26-6-27-20-23-3-9-22-11-1-29-10-24-28-17-25-21-5-2-16-19-18	79
		PSO-M	117930	-22-8-12-2-11-13-23-17-9-24-14-7-15-21-10-4-1-19-28-6-25-16-20-27-18-5-29-30-26-3	78
		En iyi çözüm	91420	23-26-19-25-20-22,11,8,9,14,27,30,12,6,28, 24,21,18,1,7,10,29,13,5,2,17,3,15,4,16)	
41	Lipa20a	PSO-St	3838	-17-8-5-4-1-3-20-2-11-14-18-16-10-7-15-19-9-12-13-6	96
		PSO(VR)	3838	-17-8-5-4-1-3-20-2-11-14-18-16-10-7-15-19-9-12-13-6	96
		PSO-M	3828	-14-7-5-13-15-12-4-2-1-20-8-17-11-9-16-18-10-3-19-6	96
		En iyi çözüm	3683		
42	Lipa20b	PSO-St	32871	-13-6-15-14-3-4-11-2-1-19-20-9-5-16-7-8-17-12-18-10	82
		PSO(VR)	32466	-1-2-6-4-5-3-7-9-14-10-11-12-8-13-19-18-15-16-20-17	83
		PSO-M	33223	-8-17-7-6-9-10-14-5-1-3-19-12-2-15-11-4-16-18-13-20	81
		En iyi çözüm	27076		

EK.4 (Devam)

43	Lipa30a	PSO-St	13620	-14-4-22-17-25-16-6-19-11-20-8-29-10-1-26-5-30-23-2-9-21-15-3-18-13-27-24-12-7-28	97
		PSO(VR)	13653	-9-5-27-8-1-21-15-3-25-14-17-18-28-30-19-12-2-29-6-4-22-24-13-23-16-10-26-20-11-7	97
		PSO-M	13620	-14-4-22-17-25-16-6-19-11-20-8-29-10-1-26-5-30-23-2-9-21-15-3-18-13-27-24-12-7-28	97
		En iyi çözüm	13178		
44	Lipa30b	PSO-St	186411	-14-27-22-7-16-18-6-20-3-25-8-13-4-23-26-19-24-30-2-1-17-15-12-29-10-21-5-11-9-28	81
		PSO(VR)	186262	-7-5-9-6-29-21-26-27-22-23-13-14-11-20-28-2-17-3-16-25-1-30-10-18-24-12-4-19-15-8	81
		PSO-M	186262	-7-5-9-6-29-21-26-27-22-23-13-14-11-20-28-2-17-3-16-25-1-30-10-18-24-12-4-19-15-8	81
		En iyi çözüm	151426		
45	Nug12	PSO-St	620	-4-8-12-2-1-9-7-3-11-5-10-6	93
		PSO(VR)	630	-2-8-12-5-1-3-7-6-10-4-11-9	92
		PSO-M	630	2-3-1-11-12-8-5-6-9-4-7-10	92
		En iyi çözüm	578	12,7,9,3,4,8,11,1,5,6,10,2	
46	Nug14	PSO-St	1130	-3-1-9-4-6-11-10-8-12-5-13-14-2-7	90
		PSO(VR)	1134	-11-6-8-13-1-5-2-3-12-10-4-14-7-9	89
		PSO-M	1110	1-2-14-11-10-6-8-3-5-13-4-9-7-12	91
		En iyi çözüm	1014	9,8,13,2,1,11,7,14,3,4,12,5,6,10	
47	Nug15	PSO-St	1304	-8-13-11-1-2-6-4-9-15-3-10-5-14-12-7	88
		PSO(VR)	1294	-1-7-11-6-5-15-9-3-2-14-8-4-12-13-10	89
		PSO-M	1266	-5-4-13-8-10-9-7-2-1-12-11-6-3-14-15	91
		En iyi çözüm	1150	(1,2,13,8,9,4,3,14,7,11,10,15,6,5,12	
48	Nug16a	PSO-St	1824	-14-16-15-1-9-10-13-12-4-8-3-11-5-7-6-2	88
		PSO(VR)	1812	-7-14-6-5-2-16-13-10-3-11-4-8-1-12-15-9	89
		PSO-M	1772	-12-10-2-15-11-1-13-7-6-3-8-14-16-5-9-4	91
		En iyi çözüm	1610	(9,14,2,15,16,3,10,12,8,11,6,5,7,1,4,13)	
49	Nug16b	PSO-St	1378	-14-8-2-7-9-10-15-1-12-11-13-6-5-16-4-3	90
		PSO(VR)	1358	-15-6-9-7-16-11-14-4-3-10-12-5-8-13-1-2	91
		PSO-M	1416	-9-8-5-16-13-11-6-15-12-7-14-2-10-1-3-4	88
		En iyi çözüm	1240	16,12,13,8,4,2,9,11,15,10,7,3,14,6,1,5	
50	Nug17	PSO-St	1980	-16-14-6-5-12-17-3-7-4-11-15-8-2-9-13-10-1	87
		PSO(VR)	1948	-6-3-17-5-13-11-12-8-14-16-10-7-9-2-1-4-15	89
		PSO-M	1940	17-1-14-6-7-15-11-13-5-12-3-8-10-4-2-9-16	89
		En iyi çözüm	1732	16,15,2,14,9,11,8,12,10,3,4,1,7,6,13,17,5	

EK.4 (Devam)

51	Nug18	PSO-St	2146	-7-16-18-1-9-10-8-3-14-12-2-13-15-17-11-6-5-4	90
		PSO(VR)	2176	-3-17-4-8-5-10-14-6-18-9-13-12-15-11-7-16-2-1	89
		PSO-M	2166	-12-8-1-9-2-16-13-17-6-3-15-14-11-5-18-10-7-4	89
		En iyi çözüm	1930	10,3,14,2,18,6,7,12,15,4,5,1,11,8,17,13,9,16	
52	Nug20	PSO-St	2922	-15-9-18-1-17-16-12-3-6-20-4-13-11-14-5-7-2-19-10-8	88
		PSO(VR)	2942	-2-18-20-17-7-1-16-12-5-15-9-14-3-10-13-4-6-11-19-8	87
		PSO-M	2950	-14-18-20-8-7-2-13-9-16-17-4-15-1-19-10-5-6-11-3-12	87
		En iyi çözüm	2570	18,14,10,3,9,4,2,12,11,16,19,15,20,8,13,17,5,7,1,6	
53	Nug21	PSO-St	2830	-15-9-11-7-1-2-17-12-4-10-5-8-16-13-14-6-19-3-18-21-20	86
		PSO(VR)	2852	-3-6-14-1-7-12-10-9-11-18-19-20-13-4-2-17-15-5-16-8-21	85
		PSO-M	2864	-12-7-9-15-6-11-17-3-19-18-2-5-20-8-1-16-13-10-4-14-21	85
		En iyi çözüm	2438	4,21,3,9,13,2,5,14,18,11,16,10,6,15,20,19,8,7,1,12,17	
54	Nug22	PSO-St	4200	-16-10-15-9-11-5-14-13-8-7-12-3-4-20-1-6-2-19-18-21-22-17	86
		PSO(VR)	3986	-20-2-16-15-1-7-6-19-4-21-9-17-5-12-22-14-11-10-18-13-3-8	90
		PSO-M	4258	-19-14-5-21-2-9-6-17-4-16-22-7-3-11-20-15-12-18-8-1-13-10	84
		En iyi çözüm	3596	2,21,9,10,7,3,1,19,8,20,17,5,13,6,12,16,11,22,18,14,15	
55	Nug24	PSO-St	4094	11-1-14-19-7-2-16-3-18-17-21-12-9-8-23-20-6-15-22-4-24-10 -13-5	85
		PSO(VR)	4040	8-6-12-23-18-4-16-14-11-7-21-22-10-24-3-5-19-1-2-20-17-9 -15-13	86
		PSO-M	4040	-8-6-12-23-18-4-16-14-11-7-21-22-10-24-3-5-19-1-2-20-17-9 -15-13	86
		En iyi çözüm	3488	(17,8,11,23,4,20,15,19,22,18,3,14,1,10,7,9,16,21,24,12,6,13,5 ,2)	
56	Nug25	PSO-St	4356	-23-7-10-15-2-8-17-4-14-3-9-1-25-13-21-22-12-5-24-19-16-18 -20-6-11	86
		PSO(VR)	4324	-21-4-10-25-2-6-17-9-13-22-14-5-20-8-1-18-15-3-11-7-23-16 -12-24-19	87
		PSO-M	4230	-25-5-3-11-7-19-22-8-13-16-15-1-20-17-10-24-2-4-18-12-23-9 -14-21-6	89
		En iyi çözüm	3744	5,11,20,15,22,2,25,8,9,1,18,16,3,6,19,24,21,14,7,10,17,12,4,23 ,13	
57	Nug27	PSO-St	6260	-16-2-13-7-19-8-1-17-5-27-21-10-25-9-3-18-20-23-4-14-24-12 -6-26-22-11-15	84
		PSO(VR)	6254	-2-16-4-6-20-11-22-8-15-21-13-19-27-10-18-23-3-26-14-5-25 -9-12-24-7-1-17	84
		PSO-M	6294	-17-23-4-11-20-10-15-27-16-19-3-18-1-26-12-2-13-21-25-5-8 -7-22-24-6-9-14	83
		En iyi çözüm	5234	(3,18,3,1,27,17,5,12,7,15,4,26,8,19,20,2,24,21,14,10,9,13,22,25 ,6,16,11	
58	Nug28	PSO-St	6176	-6-12-13-17-5-10-15-24-11-19-26-23-16-2-8-25-7-21-18-3-9-1 -28-22-14-20-4-27	84
		PSO(VR)	6154	-20-8-27-28-24-5-2-3-21-10-18-15-25-17-14-7-19-12-23-1-13-4 -9-16-11-6-26-22	84
		PSO-M	6142	-13-26-2-16-9-6-14-24-19-5-7-25-11-3-12-10-23-8-28-27-20-17 -1-18-4-21-22-15	84
		En iyi çözüm	5166	18,21,9,1,28,20,11,3,13,12,10,19,14,22,15,2,25,16,4,23,7,17,24 ,26,5,27,8,6	

EK.4 (Devam)

59	Rou12	PSO-St	260122	-5-12-6-7-3-9-10-8-11-1-2-4	91
		PSO(VR)	260122	-5-12-6-7-3-9-10-8-11-1-2-4	91
		PSO-M	261830	-12-5-2-11-10-4-8-7-3-9-6-1	90
		En iyi çözüm	235528	6,5,11,9,2,8,3,1,12,7,4,10	
60	Rou15	PSO-St	403940	-13-1-7-11-8-5-6-12-2-10-3-9-14-15-4	88
		PSO(VR)	403144	-14-2-13-6-5-8-9-1-4-3-15-11-10-12-7	88
		PSO-M	396426	-2-5-1-3-6-7-12-10-8-14-11-13-15-9-4	89
		En iyi çözüm	354210	12,6,8,13,5,3,15,2,7,1,9,10,4,14,11	
61	Rou20	PSO-St	819848	-10-3-9-19-1-14-5-20-15-12-17-13-11-18-16-4-8-2-6-7	88
		PSO(VR)	820006	-13-10-1-5-19-6-11-20-14-9-7-16-3-2-15-8-17-12-18-4	88
		PSO-M	817206	-13-4-1-5-10-18-9-12-7-2-11-16-14-17-3-8-20-6-15-19	89
		En iyi çözüm	725522	1,19,2,14,10,16,11,20,9,5,7,4,8,18,15,3,12,17,13,6	
62	Scr12	PSO-St	35894	-3-4-2-9-12-6-7-8-10-11-5-1	88
		PSO(VR)	34820	-6-4-9-1-7-2-3-10-8-5-12-11	90
		PSO-M	34650	-12-6-4-5-10-1-2-7-11-3-9-8	91
		En iyi çözüm	31410	(8,6,3,2,10,1,5,9,4,7,12,11)	
63	Scr15	PSO-St	65554	-15-11-2-4-1-12-7-6-13-14-8-9-5-10-3	78
		PSO(VR)	64754	-13-3-1-9-15-14-6-8-10-2-4-7-12-11-5	79
		PSO-M	60454	-8-7-2-1-11-5-4-6-15-14-10-12-9-13-3	85
		En iyi çözüm	51140	15,7,11,8,1,4,3,2,12,6,13,5,14,10,9	
64	Tai12a	PSO-St	247952	-7-4-12-10-2-3-6-1-5-8-9-11	91
		PSO(VR)	257410	-2-8-4-3-5-7-10-6-9-12-1-11	87
		PSO-M	248036	-8-6-11-12-9-3-7-1-2-5-4-10	90
		En iyi çözüm	224416	(8,1,6,2,11,10,3,5,9,7,12,4)	
65	Tai12b	PSO-St	41346962	-4-8-10-2-9-1-6-11-3-5-12-7	95
		PSO(VR)	45285852	-9-7-5-2-3-6-8-11-12-10-1-4	87
		PSO-M	41157477	-11-8-4-3-12-1-5-7-2-10-9-6	96
		En iyi çözüm	39464925	(9,4,6,3,11,7,12,2,8,10,1,5)	
66	Tai15a	PSO-St	427602	-9-2-8-15-1-12-3-14-6-5-13-4-7-11-10	91
		PSO(VR)	427434	-11-7-6-1-13-2-14-12-3-10-15-8-9-5-4	91
		PSO-M	427086	-9-6-12-5-8-14-2-10-3-15-7-11-4-13-1	91
		En iyi çözüm	388214	5,10,4,13,2,9,1,11,12,14,7,15,3,8,6	
67	Tai15b	PSO-St	52385462	-12-2-14-6-5-11-8-7-9-15-1-13-3-10-4	99
		PSO(VR)	52349005	-7-9-4-6-10-13-11-5-12-15-2-14-1-8-3	99
		PSO-M	52435041	-4-11-10-12-6-9-5-2-7-15-14-13-8-3-1	99
		En iyi çözüm	51765268	1,9,4,6,8,15,7,11,3,5,2,14,13,12,10	

EK.4 (Devam)

68	Tai17a	PSO-St	557784	-5-10-11-14-6-4-17-12-1-2-8-15-7-13-9-3-16	88
		PSO(VR)	550986	-14-11-16-13-17-7-3-1-2-10-5-9-4-6-15-8-12	89
		PSO-M	548188	-17-11-8-14-6-1-5-7-12-3-10-9-13-2-4-16-15	90
		En iyi çözüm	491812	12,2,6,7,4,8,14,5,11,3,16,13,17,9,1,10,15	
69	Tai20a	PSO-St	801870	-8-1-4-18-9-20-2-6-12-7-3-15-16-11-5-14-13-10-17-19	88
		PSO(VR)	809024	-10-14-13-20-6-19-8-15-2-7-17-1-18-11-16-9-4-3-5-12	87
		PSO-M	801870	-8-1-4-18-9-20-2-6-12-7-3-15-16-11-5-14-13-10-17-19	88
		En iyi çözüm	703482	10,9,12,20,19,3,14,6,17,11,5,7,15,16,18,2,4,8,13,1	
70	Tai20b	PSO-St	145030997	-10-6-4-9-2-3-1-17-5-12-15-14-20-8-13-16-11-18-7-19	84
		PSO(VR)	131275065	-11-4-2-13-20-15-8-16-14-3-9-18-7-10-12-1-6-17-5-19	93
		PSO-M	132870074	-1-15-6-12-17-2-7-11-16-14-10-18-5-13-9-4-8-20-3-19	92
		En iyi çözüm	122455319	(8,16,14,17,4,11,3,19,7,9,1,15,6,13,10,2,5,20,18,12	
71	Tai25b	PSO-St	384177050	-20-24-8-1-4-3-11-17-2-13-23-22-14-21-12-6-9-10-19-15-25-18-16-7-5	90
		PSO(VR)	420397210	-21-24-3-7-9-4-6-19-14-8-20-25-12-11-15-5-1-2-17-22-23-18-16-13-10	82
		PSO-M	419188607	-17-23-13-8-1-11-14-16-4-5-19-24-7-22-2-6-12-10-21-9-18-25-20-15-3	82
		En iyi çözüm	344355646	4,15,10,9,13,5,25,19,7,3,17,6,18,20,16,2,22,23,8,11,21,24,14,12,1	
72	Tai30a	PSO-St	2050858	26-29-15-14-28-1-5-21-13-27-24-11-19-3-17-25-8-18-9-12-22 -20-7-16-6-2-23-4-10-30	89
		PSO-VR	2067954	4-22-24-6-14-21-16-28-12-2-1-26-5-9-23-17-7-27-30-18-11-8 -25-29-20-15-10-13-19-3	88
		PSO-M	2066022	-19-12-18-4-27-5-10-7-30-17-25-21-11-16-24-2-20-9-1-29-13 -28-15-8-3-6-14-26-23-22	88
		Ro-TS	1818146		
73	Tai30b	PSO-St	826696262	-6-7-25-29-19-3-26-21-4-2-24-27-30-22-23-9-17-15-10-12-16 -1-18-8-5-28-11-14-20-13	77
		PSO(VR)	812809101	-9-17-10-21-28-19-15-13-24-23-29-1-5-27-3-4-8-18-30-6-22-11 -16-26-25-12-20-2-14-7	78
		PSO-M	804935658	-13-1-23-17-14-28-26-10-6-29-24-22-18-20-21-12-19-27-15-25 -11-7-5-2-8-4-3-9-30-16	79
		Ro-Ts	637117113		
74	Tho30	PSO-St	178832	11-3-17-27-13-29-12-14-9-18-19-7-30-23-4-26-16-8-1-6-28-24 -25-15-10-5-20-22-21-2	84
		PSO(VR)	184904	-15-28-1-25-4-29-21-5-17-26-2-27-13-10-8-24-7-12-30-6-22-9 -23-14-20-19-3-18-11-16	81
		PSO-M	185548	-1-14-10-3-2-20-13-22-17-9-12-11-8-24-19-25-15-30-6-26-28-23 -27-16-7-4-18-5-21-29	81
		En iyi çözüm	149936	(8,6,20,17,19,12,29,15,1,2,30,11,13,28,23,27,16,22,10,21,25,24 ,26,18,3,14,7,5,9,4)	

EK.4 (Devam)

75	Nug30	PSO-St	7264	-27-5-19-3-1-25-15-17-8-13-4-20-26-18-22-14-16-30-10-12-28 -24-9-21-23-7-11-29-6-2	84
		PSO(VR)	7194	-28-4-7-11-2-24-29-9-17-15-21-19-23-10-26-18-13-20-1-8-3-27 -14-22-30-25-16-12-6-5	85
		PSO-M	7224	-11-27-26-19-1-23-3-10-21-22-13-17-28-7-24-14-6-16-2-8-20-30 -4-12-5-9-29-18-25-15	85
		En iyi çözüm	6124	(14,5,28,24,1,3,16,15,10,9,21,2,4,29,25,22,13,26,17,30,6,20,19,8 ,18,7,27,12,11,23)	
76	Scr20	PSO-St	147220	-9-11-16-15-3-13-17-12-10-14-2-18-6-7-4-19-1-20-8-5	75
		PSO(VR)	143168	-11-12-16-13-14-17-10-15-3-9-6-18-19-1-7-8-5-4-2-20	77
		PSO-M	155620	-11-9-15-14-20-16-1-10-8-18-12-3-7-2-19-13-6-17-4-5	71
		En iyi çözüm	110030	20,7,12,6,4,8,3,2,14,11,18,9,19,15,16,17,13,5,10,1)	
77	Tai25a	PSO-St	1323720	-19-17-11-1-6-21-3-5-9-10-8-15-18-13-25-16-14-23-4-7-20-2-24-12-22	88
		PSO(VR)	1339356	-7-20-11-25-4-10-23-8-1-9-3-22-12-18-19-14-24-2-17-5-6-16-15-21-13	87
		PSO-M	1346980	-9-16-6-15-24-8-20-7-2-23-12-21-4-22-13-1-25-5-18-3-11-10-17-14-19	87
		Ro-Ts	1167256		

EK.5

Bölüm Sayısı	Problem	En iyi çözüme yakınsama yüzdeleri TZREST	TZREVR	TZREM	%Ort
12	Chr12a	57	66	65	62,67
	Chr12b	57	57	78	64,0
	Chr12c	78	68	66	70,7
	Had12	98	98	98	98,0
	Nug12	93	92	92	92,3
	Rou12	91	91	90	90,7
	Scr12	88	90	91	89,7
	Tai12a	91	87	90	89,3
	Tai12b	95	87	96	92,7
Ortalama		83,11	81,78	85,11	
14	Nug14	90	89	91	90,0
	Had14	98	98	98	98,0
Ortalama		94	93,5	94,5	
15	Chr15a	43	61	55	53,0
	Chr15b	42	33	40	38,3
	Chr15c	51	46	52	49,7
	Nug15	88	89	91	89,3
	Rou15	88	88	89	88,3
	Scr15	78	79	85	80,7
	Tai15a	91	91	91	91,0
	Tai15b	99	99	99	99,0
Ortalama		72,5	73,25	75,25	
16	Esc16a	92	92	94	92,7
	Esc16b	100	100	100	**100,0**
	Esc16c	98	94	92	94,7
	Esc16d	73	73	80	75,3
	Esc16e	88	88	88	88,0
	Esc16f	100	100	100	**100,0**
	Esc16g	87	81	83	83,7
	Esh16h	99	99	100	99,3
	Esc16i	100	88	100	96,0
	Esc16j	80	80	80	80,0
	Had16	97	97	97	97,0
	Nug16a	88	89	91	89,3
	Nug16b	90	91	88	89,7
Ortalama		91,69	90,15	91,77	
17	Nug17	87	89	89	88,3
	Tai17a	88	89	90	89,0
Ortalama		87,5	89	89,5	

Yakınsama Yüzdesi	Sıklık
61-70	3
71-80	0
81-90	2
91-100	4
Toplam	**9**

Yakınsama Yüzdesi	Sıklık
90-100	2

Yakınsama Yüzdesi	Sıklık
31-40	1
41-50	1
51-60	1
61-70	0
71-80	0
81-90	3
91-100	2
Toplam	**8**

Yakınsama Yüzdesi	Sıklık
71-80	2
81-90	4
91-100	7
Toplam	**13**

Yakınsama Yüzdesi	Sıklık
81-90	2

EK.5 (Devam)

Bölüm Sayısı	Problem	TZRE-ST	TZRE-VR	TZRE-M	%Ort
18	Chr18a	32	33	34	33,0
	Chr18b	64	75	66	68,3
	Had18	96	97	98	97,0
	Nug18	90	89	89	89,3
Ortalama		70,5	73,5	71,75	
19	Els19	72	74	76	74,0
20	Chr20a	45	40	47	44,0
	Chr20b	44	45	46	45,0
	Chr20c	31	32	29	30,7
	Had20	96	97	97	96,7
	Lipa20a	96	96	96	96,0
	Lipa20b	82	83	81	82,0
	Nug20	88	87	87	87,3
	Rou20	88	88	89	88,3
	Scr20	75	77	71	74,3
	Tai20a	88	87	88	87,7
	Tai20b	84	93	92	89,7
Ortalama		74,27	75,00	74,82	
21	Nug21	86	85	85	85,3
22	Chr22a	70	70	73	71,0
	Chr22b	74	72	70	72,0
	Nug22	86	90	84	86,7
Ortalama		76,67	77,33	75,67	
24	Nug24	85	86	86	85,7
25	Chr25a	35	33	35	34,3
	Nug25	86	87	89	87,3
	Tai25a	88	87	87	87,3
	Tai25b	90	82	82	84,7
Ortalama		74,75	72,25	73,25	
26	Bur26a	98	97	98	97,7
	Bur26b	97	98	98	97,7
	Bur26c	97	97	97	97,0
	Bur26d	97	97	97	97,0
	Bur26e	97	98	97	97,3
	Bur26f	97	97	97	97,0
	Bur26g	98	97	98	97,7
	Bur26h	97	97	97	97,0
Ortalama		97,25	97,31	97,33	
27	Nug27	84	84	83	83,7
28	Nug28	84	84	84	84,0

Yakınsama Yüzdesi	Sıklık
31-40	1
61-70	1
81-90	1
91-100	1
Toplam	4

Yakınsama Yüzdesi	Sıklık
71-80	1

Yakınsama Yüzdesi	Sıklık
31-40	1
41-50	2
51-60	0
61-70	0
71-80	1
81-90	5
91-100	2
Toplam	**11**

Yakınsama Yüzdesi	Sıklık
81-90	1

71-80	2
81-90	1
Toplam	**3**

81-90	1

31-40	1
81-90	3
Toplam	**4**

Yakınsama Yüzdesi	Sıklık
91-100	8

Yakınsama Yüzdesi	Sıklık
81-90	1

81-90	1

EK.5 (Devam)

Bölüm Sayısı	Problem	TZRE-ST	TZRE-VR	TZRE-M	%Ort
	Kra30a	77	78	80	78,3
	Kra30b	77	79	78	78,0
30	Lipa30a	97	97	97	97,0
	Lipa30b	81	81	81	81,0
	Nug30	84	85	85	84,7
	Tai30a	89	88	88	88,3
	Tai30b	77	78	79	78,0
	Tho30	84	81	81	82,0
Ortalama		83,3	83,4	83,6	

Yakınsama Yüzdesi	Sıklık
71-80	3
81-90	4
91-100	1
Toplam	**8**

EK.6

```
MODEL:

!nug25 %10;

 SETS:
   FLOW/1..25/;   !  There are twelwe departments;
   DISTANCE/1..25/;     !  There are twelwe departments;
   FXD( FLOW, DISTANCE):   X;  ! Flow to distance assignment vbl;
   DXD( DISTANCE, DISTANCE):     T; ! Distance between departments;
   FXF( FLOW, FLOW): N; ! No. transfers between departments;
 ENDSETS
 DATA:

      N =    0 1 2 3 4 1 2 3 4 5 2 3 4 5 6 3 4 5 6 7 4 5 6 7 8
             1 0 1 2 3 2 1 2 3 4 3 2 3 4 5 4 3 4 5 6 5 4 5 6 7
             2 1 0 1 2 3 2 1 2 3 4 3 2 3 4 5 4 3 4 5 6 5 4 5 6
             3 2 1 0 1 4 3 2 1 2 5 4 3 2 3 6 5 4 3 4 7 6 5 4 5
             4 3 2 1 0 5 4 3 2 1 6 5 4 3 2 7 6 5 4 3 8 7 6 5 4
             1 2 3 4 5 0 1 2 3 4 1 2 3 4 5 2 3 4 5 6 3 4 5 6 7
             2 1 2 3 4 1 0 1 2 3 2 1 2 3 4 3 2 3 4 5 4 3 4 5 6
             3 2 1 2 3 2 1 0 1 2 3 2 1 2 3 4 3 2 3 4 5 4 3 4 5
             4 3 2 1 2 3 2 1 0 1 4 3 2 1 2 5 4 3 2 3 6 5 4 3 4
             5 4 3 2 1 4 3 2 1 0 5 4 3 2 1 6 5 4 3 2 7 6 5 4 3
             2 3 4 5 6 1 2 3 4 5 0 1 2 3 4 1 2 3 4 5 2 3 4 5 6
             3 2 3 4 5 2 1 2 3 4 1 0 1 2 3 2 1 2 3 4 3 2 3 4 5
             4 3 2 3 4 3 2 1 2 3 2 1 0 1 2 3 2 1 2 3 4 3 2 3 4
             5 4 3 2 3 4 3 2 1 2 3 2 1 0 1 4 3 2 1 2 5 4 3 2 3
             6 5 4 3 2 5 4 3 2 1 4 3 2 1 0 5 4 3 2 1 6 5 4 3 2
             3 4 5 6 7 2 3 4 5 6 1 2 3 4 5 0 1 2 3 4 1 2 3 4 5
             4 3 4 5 6 3 2 3 4 5 2 1 2 3 4 1 0 1 2 3 2 1 2 3 4
             5 4 3 4 5 4 3 2 3 4 3 2 1 2 3 2 1 0 1 2 3 2 1 2 3
             6 5 4 3 4 5 4 3 2 3 4 3 2 1 2 3 2 1 0 1 4 3 2 1 2
             7 6 5 4 3 6 5 4 3 2 5 4 3 2 1 4 3 2 1 0 5 4 3 2 1
             4 5 6 7 8 3 4 5 6 7 2 3 4 5 6 1 2 3 4 5 0 1 2 3 4
             5 4 5 6 7 4 3 4 5 6 3 2 3 4 5 2 1 2 3 4 1 0 1 2 3
             6 5 4 5 6 5 4 3 4 5 4 3 2 3 4 3 2 1 2 3 2 1 0 1 2
             7 6 5 4 5 6 5 4 3 4 5 4 3 2 3 4 3 2 1 2 3 2 1 0 1
             8 7 6 5 4 7 6 5 4 3 6 5 4 3 2 5 4 3 2 1 4 3 2 1 0
;  ! No. transfers between departments;


      T =  0  3  2  0  0 10  5  0  5  2  0  0  2  0  5  3  0  1 10  0
2  1  1  1  0
 3  0  4  0 10  0  0  2  2  1  5  0  0  0  0  0  1  6  1  0  2  2  5
1 10
 2  4  0  3  4  5  5  5  1  4  0  4  0  4  0  3  2  5  5  2  0  0  3
1  0
 0  0  3  0  0  0  2  2  0  6  2  5  2  5  1  1  1  2  2  4  2  0  2
2  5
 0 10  4  0  0  2  0  0  0  0  0  0  0  0  2  0  0  2  0  5  0  2  1
0  2
10  0  5  0  2  0 10 10  5 10  6  0  0 10  2 10  1  5  5  2  5  0  2
0  1
 5  0  5  2  0 10  0  1  3  5  0  0  2  4  5 10  6  0  5  5  5  0  5
5  0
```

EK.6 (Devam)

```
 0  2  5  2  0 10  1  0 10  2  5  2  0  3  0  0  0  4  0  5  0  5  2
2  5
 5  2  1  0  0  5  3 10  0  5  6  0  1  5  5  5  2  3  5  0  2 10 10
1  5
 2  1  4  6  0 10  5  2  5  0  0  1  2  1  0  0  0  0  6  6  4  5  3
2  2
 0  5  0  2  0  6  0  5  6  0  0  2  0  4  2  1  0  6  2  1  5  0  0
1  5
 0  0  4  5  0  0  0  2  0  1  2  0  2  1  0  3 10  0  0  4  0  0  4
2  5
 2  0  0  2  0  0  2  0  1  2  0  2  0  4  5  0  1  0  5  0  0  0  5
1  1
 0  0  4  5  0 10  4  3  5  1  4  1  4  0  0  0  2  2  0  2  5  0  5
2  5
 5  0  0  1  2  2  5  0  5  0  2  0  5  0  0  2  0  0  0  6  3  5  0
0  5
 3  0  3  1  0 10 10  0  5  0  1  3  0  0  2  0  0  5  5  1  5  2  1
2 10
 0  1  2  1  0  1  6  0  2  0  0 10  1  2  0  0  0  5  2  1  1  5  6
5  5
 1  6  5  2  2  5  0  4  3  0  6  0  0  2  0  5  5  0  4  0  0  0  0
5  0
10  1  5  2  0  5  5  0  5  6  2  0  5  0  0  5  2  4  0  5  4  4  5
0  2
 0  0  2  4  5  2  5  5  0  6  1  4  0  2  6  1  1  0  5  0  4  4  1
0  2
 2  2  0  2  0  5  5  0  2  4  5  0  0  5  3  5  1  0  4  4  0  1  0
10  1
 1  2  0  0  2  0  0  5 10  5  0  0  0  0  5  2  5  0  4  4  1  0  0
0  0
 1  5  3  2  1  2  5  2 10  3  0  4  5  5  0  1  6  0  5  1  0  0  0
0  0
 1  1  1  2  0  0  5  2  1  2  1  2  1  2  0  2  5  5  0  0 10  0  0
0  2
 0 10  0  5  2  1  0  5  5  2  5  5  1  5  5 10  5  0  2  2  1  0  0
2  0
;

  ! distance between locations;

 ENDDATA

 SETS:
   ! Transfer between 2 departments must be required and related
     to 2 different locations. Warning: this set gets big fast. ;
   TGTG( FLOW, DISTANCE, FLOW, DISTANCE)| &1 #LT# &3 #AND#
          (( N( &1, &3) #NE# 0) #AND# ( T( &2, &4) #NE# 0)
      #OR# ( N( &3, &1) #NE# 0) #AND# ( T( &4, &2) #NE# 0)): Y;
 ENDSETS

 ! Each department, B, must be assigned to a location;
  @FOR( FLOW( B):
        @SUM( DISTANCE( J): X( B, J)) = 1; );
```

EK.6 (Devam)

```
 ! Each gate, J, can receive at most one department;
  @FOR( DISTANCE( J):
        @SUM( FLOW( B): X( B, J)) <= 1; );

 ! Force Y(B,J,C,K)=1 if B assigned to J and C assigned to K;
 ! Assumes the T() and N() matrices are nonnegative;
  @FOR( TGTG( B, J, C, K):
       Y( B, J, C, K) >= X( B, J) + X( C, K) - 1; );

 ! Min the sum of transfers * distance;
  MIN = @SUM( TGTG( B, J, C, K): Y( B, J, C, K) *
        ( N( B, C) * T( J, K) + N( C, B) * T( K, J)) ;);

!Department Restrictions;
X(1,15)=0;
X(1,16)=0;
X(2,12)=0;
X(2,20)=0;
X(3,8)=0;
X(3,13)=0;
X(4,3)=0;
X(4,20)=0;
X(5,17)=0;
X(5,19)=0;
X(6,12)=0;
X(6,20)=0;
X(7,6)=0;
X(7,19)=0;
X(8,3)=0;
X(8,18)=0;
X(9,17)=0;
X(9,19)=0;
X(10,7)=0;
X(10,11)=0;
X(11,3)=0;
X(11,17)=0;
X(12,8)=0;
X(12,22)=0;
X(13,4)=0;
X(13,22)=0;
X(14,12)=0;
X(14,16)=0;
X(15,14)=0;
X(15,23)=0;
X(16,18)=0;
X(16,22)=0;
X(17,8)=0;
X(17,17)=0;
X(18,9)=0;
X(18,14)=0;
X(19,3)=0;
X(19,23)=0;
X(20,4)=0;
X(20,16)=0;
```

EK.6 (Devam)

```
X(21,5)=0;
X(21,10)=0;
X(22,1)=0;
X(22,3)=0;
X(23,6)=0;
X(23,11)=0;
X(24,8)=0;
X(24,14)=0;
X(25,3)=0;
X(25,4)=0;
  ! Make the X's binary;
  @FOR( FXD: @BIN( X););
 END
```

EK.6 (Devam)

```
Global optimal solution found at step:       1070707
 Objective value:                            12470.00
 Branch count:                                  16902

                     Variable           Value        Reduced Cost
                      X( 1, 1)       0.0000000            6120.000
                      X( 1, 2)       0.0000000            14760.00
                      X( 1, 3)       0.0000000            12960.00
                      X( 1, 4)       0.0000000           0.0000000
                      X( 1, 5)       0.0000000            6400.000
                      X( 1, 6)       0.0000000            11340.00
                      X( 1, 7)       0.0000000            15120.00
                      X( 1, 8)       0.0000000            680.0000
                      X( 1, 9)       0.0000000           0.0000000
                     X( 1, 10)        1.000000            96600.00
                     X( 1, 11)       0.0000000           0.0000000
                     X( 1, 12)       0.0000000            16300.00
                      X( 2, 1)       0.0000000            4256.000
                      X( 2, 2)       0.0000000            9530.000
                      X( 2, 3)       0.0000000           0.0000000
                      X( 2, 4)       0.0000000            8640.000
                      X( 2, 5)       0.0000000            7158.000
                      X( 2, 6)       0.0000000            10272.00
                      X( 2, 7)       0.0000000            4096.000
                      X( 2, 8)        1.000000            121136.0
                      X( 2, 9)       0.0000000            13222.00
                     X( 2, 10)       0.0000000            2208.000
                     X( 2, 11)       0.0000000           0.0000000
                     X( 2, 12)       0.0000000           0.0000000
                      X( 3, 1)        1.000000            55862.00
                      X( 3, 2)       0.0000000            3460.000
                      X( 3, 3)       0.0000000           0.0000000
                      X( 3, 4)       0.0000000            4916.000
                      X( 3, 5)       0.0000000            750.0000
                      X( 3, 6)       0.0000000            9336.000
                      X( 3, 7)       0.0000000           0.0000000
                      X( 3, 8)       0.0000000            7378.000
                      X( 3, 9)       0.0000000           0.0000000
                     X( 3, 10)       0.0000000            946.0000
                     X( 3, 11)       0.0000000            6216.000
                     X( 3, 12)       0.0000000            4902.000
                      X( 4, 1)       0.0000000            14236.00
                      X( 4, 2)       0.0000000           0.0000000
                      X( 4, 3)       0.0000000            20108.00
                      X( 4, 4)        1.000000            113442.0
                      X( 4, 5)       0.0000000            12598.00
                      X( 4, 6)       0.0000000            2530.000
                      X( 4, 7)       0.0000000            20714.00
                      X( 4, 8)       0.0000000            9680.000
                      X( 4, 9)       0.0000000            8712.000
                     X( 4, 10)       0.0000000           0.0000000
                     X( 4, 11)       0.0000000            690.0000
                     X( 4, 12)       0.0000000           0.0000000
                      X( 5, 1)       0.0000000           0.0000000
                      X( 5, 2)       0.0000000            3096.000
```