



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**TESİS YERLEŞİMİ DÜZENLENMESİ PROBLEMİ
İÇİN BİR TABU ARAMA SEZGİSEL
ALGORİTMASI**

**Ayşe Nur ADIGÜZEL
Endüstri Mühendisliği Anabilim Dalı**

**Danışman
Prof.Dr. Şakir ESNAF**

Nisan, 2012

İSTANBUL



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**TESİS YERLEŞİMİ DÜZENLENMESİ PROBLEMİ
İÇİN BİR TABU ARAMA SEZGİSEL
ALGORİTMASI**

**Ayşe Nur ADIGÜZEL
Endüstri Mühendisliği Anabilim Dalı**

**Danışman
Prof.Dr. Şakir ESNAF**

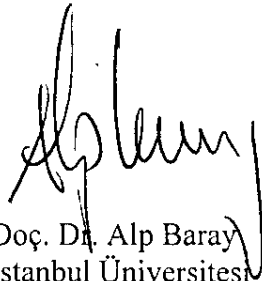
Nisan, 2012


İSTANBUL


Bu çalışma 16/ 05 / 2012 tarihinde aşağıdaki jüri tarafından Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi


Prof. Dr. Şakir ESNAF (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi


Doç. Dr. Alp Baray
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Necdet OZÇAKAR
İstanbul Üniversitesi
İşletme Fakültesi


Doç. Dr. Mehpare TİMOR
İstanbul Üniversitesi
İşletme Fakültesi



Yrd. Doç. Dr. Tark KÜÇÜKDENİZ
İstanbul Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Bilinçli tüketimin artması ve kaynakların azalması, gerek kamu gerek özel sektördeki işletmelerin ayakta kalabilme ve gelişebilmeleri için kendilerini yenilemeleri, verimli üretim yapmaları, maliyetlerini minimum düzeye indirmeleri, ürün bileşimindeki değişimlere ve taleplere çabuk tepki vererek rekabet ortamına ve sürekli değişen teknolojiye ayak uydurmaları gerektiği gerçeğini ön plana çıkarmıştır. Gelişen teknoloji ile pazar gereksinimleri hızla değişmektedir. Rekabetçi ortamda ayakta kalabilmek ve böyle bir çevrede verimli çalışabilmek için üretim şirketlerinin tesis yerleşimlerini maliyetlerini minimum düzeye indirecek şekilde düzenlenmeleri ve piyasa değişimine uydurmaları gerekmektedir.

Bu çalışmada, taşıma maliyetini minimum kılmayı ve üretim verimliliğini artırmayı amaçlayan tesis yerleşim probleminin çözümü için gelişmiş bir meta-sezgisel yöntem olan tabu arama metodundan faydalanılmıştır.

Tez çalışmamda bana yardımcı olan ve desteklerini esirgemeyen tez danışmanın Sayın Prof. Dr. Şakir Esnaf'a tezimin bu aşamaya gelmesinde büyük katkısı olduğu ve beni yönlendirdiği için sonsuz teşekkürlerimi sunarım.

Hayatım boyunca tüm kararlarımda bana güvenip beni destekleyen, bana karşı olan sevgi ve saygılarını her zaman hissettiğim ve her konuda olduğu gibi bu çalışmam süresince desteklerini esirgemeyen başta babam olmak üzere tüm aileme ve arkadaşlarıma gösterdikleri sabır ve özveriden dolayı şükranlarımı sunarım.

Nisan, 2011

Ayşe Nur Adıgüzel

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ.....	vi
TABLO LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET.....	ix
SUMMARY	x
1. GİRİŞ	1
2. GENEL KISIMLAR	2
2.1. TESİS YERLEŞİMİ DÜZENLENMESİNİN TEMELLERİ.....	2
2.1.1. Tesis Yerleşimi Düzenlenmesinin Önemi.....	2
2.1.2. Tesis Yerleşim Düzenlenmesinin Amacı	3
2.1.3. Tesis Yerleşimi Düzenlenmesinin Etmenleri	5
2.1.4. Tesis Yerleşimi Düzenlenmesi Probleminin Gereksinim Duyduğu Veriler	5
2.1.5. Tesis Yerleşim Düzenlenmesinin İlkeleri.....	6
2.1.6. Genel Akış Modelleri.....	7
2.1.6.1. İş istasyonları İçindeki Akış	7
2.1.6.2. Bölümler İçindeki Akış	7
2.1.6.3. Bölümler Arasındaki Akış	8
2.1.7. İş Akış Tipleri.....	9
2.1.7.1. Akış Sistemini Etkileyen Öğeler	9
2.1.7.2. İş Akış Modelleri	10
2.1.8. Tesis Yerleşim Tipleri.....	14
2.1.9. Tesis Yerleşimi Düzenlenmesi Problemlerinin Sınıflandırılması	17
2.2. TESİS YERLEŞİMİ DÜZENLENMESİ İÇİN GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİ.....	18
2.2.1. Geleneksel Yaklaşımlar	18
2.2.1.1. Tasarım Problemleri.....	21
2.2.1.2. Gezi Şeması.....	25
2.2.1.3. Atama yöntemi	25

2.2.1.4. <i>Wimmert Yöntemi</i>	25
2.2.1.5. <i>Spiral Analiz Yöntemi</i>	25
2.2.1.6. <i>Hat Dengeleme</i>	26
2.2.1.7. <i>Doğrusal Hat Yerleştirme Yöntemi</i>	26
2.2.1.8. <i>Dal-Sınır Algoritması</i>	26
2.2.1.9. <i>Montreuil's Karma-Tamsayı Programlama Algoritması</i>	26
2.2.1.10. <i>Grafik Teorisi Yaklaşımı</i>	27
2.2.1.11. <i>Karesel Atama Problemi</i>	27
2.2.2. Sezgisel Yaklaşımlar	28
2.2.3. Meta Sezgisel Yaklaşımlar	30
2.2.3.1. <i>Değişken Komşuluk Arama</i>	31
2.2.3.2. <i>Parçacık Sürü Optimizasyonu Algoritması</i>	31
2.2.3.3. <i>Karınca Kolonisi Optimizasyonu Algoritması</i>	33
2.2.3.4. <i>Genetik Algoritma</i>	34
2.2.3.5. <i>Yapay Sinir Ağları</i>	35
2.2.3.6. <i>Tavlama Benzetimi Algoritması</i>	36
2.2.3.7. <i>Tepe Tırmanma Algoritması</i>	37
2.2.3.8. <i>Tabu Arama Algoritması</i>	38
2.3. BİLGİSAYAR DESTEKLİ TESİS PLANLAMASI	39
2.3.1. Tesis Yerleşimi Düzenlenmesi Problemi İçin Geliştirilen Temel Algoritmalar	39
2.3.1.1. <i>Kuruluş algoritmaları</i>	39
2.3.1.2. <i>Geliştirme algoritmaları</i>	42
2.3.1.3. <i>Melez algoritmalar</i>	43
3. MALZEME VE YÖNTEM	44
3.1. TABU ARAMA ALGORİTMASININ ORTAYA ÇIKIŞI	44
3.2. TABU ARAMA ALGORİTMASINA GENEL BAKIŞ	45
3.2.1. Komşu Arama Algoritması	50
3.2.2. Temel Tabu Arama Algoritması	51
3.2.3. Problem Formülasyonu	52
3.2.4. Tabu Aramanın Temel Kavramları	56
3.2.5. Seçim Kurallarının Değiştirilmesi	64
3.2.6. Tabu Arama Parametreleri	65
3.2.7. Tabu Aramanın Temel Stratejileri	66
3.2.7.1. <i>Riskli (yasaklama) strateji</i>	66
3.2.7.2. <i>Aspirasyon kriteri (Serbest strateji)</i>	66
3.2.7.3. <i>Klasik seçme stratejisi</i>	67
3.2.7.4. <i>İlk ve en iyiyi seçme stratejisi</i>	67
3.2.7.5. <i>Öğrenme stratejisi</i>	67

3.2.7.6. Yoğunlaşma (kuvvetlendirme) stratejisi.....	67
3.2.7.7. Çeşitleme stratejisi.....	68
3.2.7.8. Stratejik Salınım.....	68
3.2.7.9. Rota Birleştirme.....	68
3.2.7.10. Elit Durumlar.....	69
3.2.7.11. Yeniden Başlama Stratejisi.....	69
3.2.7.12. Kritik Olay Hafızası.....	70
3.2.8. Tabu Arama Bellek Yapıları.....	71
3.2.8.1. Yakın Geçmişe Dayalı Bellek Yapısı.....	72
3.2.8.2. Sıklığa Dayalı Bellek Yapısı.....	73
3.2.8.3. Karma Bellek Yapısı.....	75
3.2.9. Programlama ve Sunumun Etkisi.....	75
3.2.10. Tabu Arama Algoritmasında Hafıza Kullanımı.....	76
3.2.10.1. Kısa Dönem Hafıza.....	76
3.2.10.2. Orta dönem hafıza.....	77
3.2.10.3. Uzun Dönem Hafıza.....	78
3.2.11. Tabu Aramanın Diğer Yöntemlerle Karşılaştırılması.....	80
3.2.12. Tabu Arama Uygulamaları.....	80
3.3. KARESEL ATAMA PROBLEMİ.....	80
3.3.1. KAP Tabanlı Tesis Yerleşimi Problemleriyle İlgili Çalışmalar.....	85
3.3.2. Açıklayıcı Basit Bir Problem ve Çözümü.....	86
4. BULGULAR.....	91
4.1. PROBLEMLE İLGİLİ VERİLER.....	91
4.1.1. NUG12 Problemi için Elde Edilen Sonuçlar.....	95
4.1.2. NUG14 Problemi için Elde Edilen Sonuçlar.....	96
4.1.3. NUG15 Problemi için Elde Edilen Sonuçlar.....	97
4.1.4. NUG16a Problemi için Elde Edilen Sonuçlar.....	98
4.1.5. NUG16b Problemi için Elde Edilen Sonuçlar.....	99
4.1.6. NUG17 Problemi için Elde Edilen Sonuçlar.....	100
4.1.7. NUG18 Problemi için Elde Edilen Sonuçlar.....	101
4.1.8. NUG20 Problemi için Elde Edilen Sonuçlar.....	102
4.1.9. NUG21 Problemi için Elde Edilen Sonuçlar.....	103
4.1.10. NUG22 Problemi için Elde Edilen Sonuçlar.....	104
4.1.11. NUG24 Problemi için Elde Edilen Sonuçlar.....	105
4.1.12. NUG25 Problemi için Elde Edilen Sonuçlar.....	106
4.1.13. NUG27 Problemi için Elde Edilen Sonuçlar.....	107
4.1.14. NUG28 Problemi için Elde Edilen Sonuçlar.....	108
4.1.15. NUG30 Problemi için Elde Edilen Sonuçlar.....	109
4.1.16. Farklı Tabu Süreleri için Sonuçların Karşılaştırılması.....	110

4.1.17. Farklı İterasyon Sayıları için Sonuçların Karşılaştırılması	113
5. TARTIŞMA VE SONUÇ.....	117
KAYNAKLAR	121
ÖZGEÇMİŞ.....	128

ŞEKİL LİSTESİ

Şekil 2.1 : Ürün Akışı Modelleri.....	7
Şekil 2.2 : İş İstasyonu-Koridor Yerleşimine Göre Akış Modelleri	8
Şekil 2.3 : Genel Akış Modelleri	8
Şekil 2.4 : İş Akış Modelleri	12
Şekil 2.5 : Yatay İş Akışı Hatları	13
Şekil 2.6 : Klasik Optimizasyon Yöntemleri	20
Şekil 2.7 : Sistematik tesis düzenlenme yordamı.....	23
Şekil 3.1 : Temel Tabu Arama Algoritması.....	49
Şekil 3.2 : Tabu Aramada mevcut çözümden komşu çözümlere geçiş.....	53
Şekil 3.3 : Örnek yakınsama grafiği.....	54
Şekil 3.4 : Mevcut bir durumun komşuluğu	55
Şekil 3.5 : Uygun bir durumdan uygun olmayan bir duruma geçiş	55
Şekil 3.6 : Uygun olmayan bir durumdan uygun olan bir duruma geçiş	56
Şekil 3.7 : Tabu Arama Algoritmasının Temel Akış Şeması.....	57
Şekil 3.8 : Tabu Aramanın kullanıldığı bir arama örneği	75
Şekil 3.9 :KAP çözümü için örnek bir yerleşim	88
Şekil 4.1 :12 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	96
Şekil 4.2: 14 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	97
Şekil 4.3: 15 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	98
Şekil 4.4: 16 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması (a).....	99
Şekil 4.5: 16 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması (b) ...	100
Şekil 4.6: 17 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	101
Şekil 4.7: 18 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	102
Şekil 4.8: 20 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	103
Şekil 4.9: 21 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	104
Şekil 4.10: 22 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	105
Şekil 4.11: 24 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	106
Şekil 4.12: 25 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	107
Şekil 4.13: 27 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	108
Şekil 4.14: 28 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	109
Şekil 4.15: 30 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması	110
Şekil 4.16: Tabu Sürelerine Göre Maliyetlerin Karşılaştırılması.....	112
Şekil 4.17: Tabu Sürelerine Göre Maliyetten Sapma Miktarlarının Karşılaştırılması	113
Şekil 4.18: İterasyon Sayılarına Göre Maliyetlerin Karşılaştırılması	115
Şekil 4.19: İterasyon Sayılarına Göre Maliyetten Sapma Miktarlarının Karşılaştırılması.....	115
Şekil 5.1 : Bulunan En İyi Maliyet Değerlerinin Optimum Değerlerden Sapma Miktarları	118

TABLO LİSTESİ

Tablo 4.1: NUG Problemleri için Literatürde Bulunan Optimum Sonuçlar	95
Tablo 4.2: 12 Departman için Bulunan Çözümler	96
Tablo 4.3: 14 Departman için Bulunan Çözümler	97
Tablo 4.4: 15 Departman için Bulunan Çözümler	98
Tablo 4.5: 16-a Departman için Bulunan Çözümler	99
Tablo 4.6: 16-b Departman için Bulunan Çözümler	100
Tablo 4.7: 17 Departman için Bulunan Çözümler	101
Tablo 4.8: 18 Departman için Bulunan Çözümler	102
Tablo 4.9: 20 Departman için Bulunan Çözümler	103
Tablo 4.10: 21 Departman için Bulunan Çözümler	104
Tablo 4.11: 22 Departman için Bulunan Çözümler	105
Tablo 4.12: 24 Departman için Bulunan Çözümler	106
Tablo 4.13: 25 Departman için Bulunan Çözümler	107
Tablo 4.14: 27 Departman için Bulunan Çözümler	108
Tablo 4.15: 28 Departman için Bulunan Çözümler	109
Tablo 4.16: 30 Departman için Bulunan Çözümler	110
Tablo 4.17: Tabu süresi=5 için Elde Edilen En İyi Maliyet Değerleri	111
Tablo 4.18: Tabu süresi=25 için Elde Edilen En İyi Maliyet Değerleri	111
Tablo 4.19: Tabu süresi=100 için Elde Edilen En İyi Maliyet Değerleri	112
Tablo 4.20: İterasyon Sayısı=500 için Elde Edilen En İyi Maliyet Değerleri	113
Tablo 4.21: İterasyon Sayısı=1000 için Elde Edilen En İyi Maliyet Değerleri	114
Tablo 4.22: İterasyon Sayısı=5000 için Elde Edilen En İyi Maliyet Değerleri	114
Tablo 5.1 : NUG Problemleri için Literatürdeki En iyi Değerler ve TA ile Elde Edilen En İyi Maliyet Değerlerinin Karşılaştırılması	119
Tablo 5.2 : İterasyon Sayısına göre Test İstatistikleri.....	114
Tablo 5.3 : Tabu Süresine göre Test İstatistikleri	114

SEMBOL LİSTESİ

f_{ij}	: i tesisinden j tesisine olan is akısı
c_{jl}	: j yerlesiminden 1 yerlesimine birim yük tasıma maliyeti
d_{ij}	: i ve j tesislerine atanmış alanların merkezleri arasındaki dikdörtgensel uzaklığı
$f(x)$: Amaç fonksiyonu
$N(x)$: x çözümünün komşuluğu
x	: Uygun bir çözüm
DKA	: Değişken komşuluk arama
GA	: Genetik algoritma
KKO	: Karınca kolonisi optimizasyonu
KSO	: Kuş sürüsü optimizasyonu
Q	: Miktar
P	: Ürün
PSO	: Parçacık sürü optimizasyonu
R	: Rotalama
S	: Hizmetler
STD	: Standart tesis düzenlenmesi
T	: Zamanlama
TA	: Tabu Arama
TB	: Tavlama Benzetimi
TT	: Tepe tırmanma
TYDP	: Tesis yerleşimi düzenlenmesi problemi
YSA	: Yapay sinir ağları

ÖZET

TESİS YERLEŞİMİ DÜZENLENMESİ PROBLEMİ İÇİN BİR TABU ARAMA SEZGİSEL ALGORİTMASI

Üretim araçlarının, yardımcı tesislerin veya iş istasyonlarının ve taşıma, depolama, kalite kontrolü gibi üretimle ilgili faaliyetlerin fiziksel konumları açısından bir bütün olarak koordinasyonuna tesis yerleşimi düzenlenmesi denir.

Tesis yerleşimi düzenlenmesi problemi tümleşik yapısı nedeniyle sadece küçük çaplı problemlerde mevcut teknikler kullanılarak kabul edilebilir bir sürede çözülebilir. Bunun sonucu olarak, tasarlanan problem için çeşitli sezgisel yöntemler geliştirilmiştir.

Bu çalışmada, optimum çözümü elde etmek için çözüm uzayını hareket mekanizmasıyla araştıran ve sahip olduğu yapay hafıza özelliği ile önceki tasarımları üretmeyerek yerel optimumlardan kurtulan, yani; son çözüme götüren adımın döngüsel hareketler yaratmasını engellemek için sonraki adımlarda tekrarını yasaklayan veya cezalandıran gelişmiş bir meta-sezgisel yöntem olan tabu arama metodu kullanılarak, tesis yerleşimi düzenlenmesi probleminin çözümüne yönelik etkin bir yöntem tasarlanmış ve yöntem literatürdeki bazı test problemlerine uygulanmıştır. Elde edilen sonuçlar literatürde yer alan problemlerin çözümü ile bulunan en iyi sonuçlarla karşılaştırılmıştır.

SUMMARY

A TABU SEARCH HEURISTIC ALGORITHM FOR A FACILITY LAYOUT PROBLEM

Layout arrangement within facility is described as the whole coordination of applications related to production such as the transportation, storage, and quality control of the production tools, sub-facilities or work stations in terms of their physical conditions.

The problem of facility layout, due to its integrated structure, can only be solved by using available techniques in an acceptable period on small-scale problems. As a result heuristic methods have been developed for the designed problems.

In this study, in order to get optimum solutions of the literature benchmark problems, it is aimed to design an effective method for the facility layout problem, tabu search method which is a meta-heuristic method, searching solution space with movement mechanism and due to its existing artificial memory feature, getting rid of the local optimums without producing the previous designs, that is, forbidding or punishing the repetition in next steps in order to prevent the last step into the result from making cyclic movements. The obtained solutions are compared with the best known solutions.

1. GİRİŞ

Tesis yerleşimi düzenlenmesi problemi, araç-gereçleri, departmanları veya diğer kaynakları belirli bir amaç fonksiyonu çerçevesinde kısıtlar çakışmayacak ve optimum performans sağlayacak şekilde yerleştirme kararını içermektedir. Her ürünün kendine özgü gereksinimleri olabilir ve tüm ürünler için toplam üretim maliyetinin en az olması sağlanacak şekilde yerleşim düzenlenmesi yapılmalıdır. Yerleşim kararları hızlı ve doğru verilmelidir. Çünkü doğru alınamayan kararlar üretim esnasında ortaya çıkmakta ve bu da beklenmedik maliyetlere yol açmaktadır. Genellikle yeterli zaman ve kaynak olmadığı için yeni bir üretim tesisi inşa etmektense mevcut tesisin analiz edilmesi ve yerleşimin gerekli değişikliklere göre yeniden tasarlanması gerekmektedir. Tesiste etkin bir yerleşim düzenlenmesi işletmelerin tüm verimliliğine katkıda bulunurken etkin olmayan bir yerleşim düzenlenmesi ise işlem sürecindeki stokların birikmesine, malzeme taşıma sistemlerinin aşırı yüklenilmesine ve uzun kuyruklara neden olur.

Tesis yerleşimi, zaman ve maliyet bakımından üretim verimliliğine büyük katkısı olan anahtar alanlardan biri, üretim sistemleri tasarım sürecinin ayrılmaz bir parçasıdır. Tedarik süresi, hammadde/işlem hacim oranı ve iş süreci ile ölçülen çalışma performansını doğrudan etkiler. Bu nedenle tesis yerleşimi düzenlenmesi probleminde dikkate alınan en yaygın amaç, nicel faktörler olan malzeme taşıma ve yeniden yerleşim maliyetini en aza indirerek mevcut tesisleri en az maliyetli alanlara yerleştirme planıdır. Malzeme taşıma maliyetleri, tesislerin konumları arasındaki uzaklık ve tesisler arası malzeme akışı miktarına dayanılarak saptanır. Bunun dışında tesis güvenliği, malzeme tasarım değişikliği için yerleşim esnekliği, toz, gürültü ve estetik görünüm gibi nitel faktörler de tesis yerleşimi düzenlenmesinde göz önünde bulundurulabilir.

2. GENEL KISIMLAR

2.1. TESİS YERLEŞİMİ DÜZENLENMESİNİN TEMELLERİ

2.1.1. Tesis Yerleşimi Düzenlenmesinin Önemi

Hammaddenin ürüne dönüşmesi için, bir dizi işlem görmesi gerekir. Tesisler, bu işlemlerin gerçekleştirildiği yerlerdir. Tesis, bir tezgah, bir iş merkezi, bir atölye, vb. olabilir. Parça her tesiste bir veya birkaç işlem görür, tek başına veya partiler halinde bir sonraki işlemi görmek üzere sırası gelen tesise gider. İmalatta her parça kendine özgü bir işlem sırası izler. Üretilcek parça çeşidi ve miktarı talebe bağlıdır ve bu veriler tesis planlamacısının kontrolü dışındadır. Taşıma (malzeme aktarma) maliyetlerinin hesabı, taşıma sayılarına dayanmaktadır. Bu sayılar da, ürün karması (hangi parçadan kaç adet üretildiği), rotalar ve parti büyüklüklerine bağlıdır. Taşıma maliyetleri, günlük, haftalık, aylık veya yıllık dönemlere göre hesaplanabilir. Bu maliyetler, taşıma sayıları dışında, (yükleme ve boşaltma işlemlerinden kaynaklanan) sabit maliyetlere, (bir yükü bir metre taşımanın maliyeti olan) birim taşıma maliyetlerine ve en önemlisi de tesisler arası uzaklıklara bağlıdır (İşlier, 2000).

Tesis yerleşimi düzenlenmesinin asıl amacı tesiste yer alan üretim faaliyetlerindeki canlı ve cansız varlıkların tümünün hareket miktarlarını minimum düzeye indirmektir. Yerleşim düzeninin etkin olmaması malzeme taşıma maliyetini ve sabit tesis maliyetlerini yükseltir. Bundan da önemlisi, etkin olmayan yerleşimin; enerji kaybı, kargaşa, yüksek ıskarta oranı, gecikme, kontrol ve yönetim güçlüğü gibi üretimle beraber süren ve maliyetleri olumsuz yönde etkileyen bir neden olmasıdır. Etkin olmayan yerleşim düzeni fabrikanın üretim kapasitesinden yararlanma oranını düşürür.

Tesis yerleşimi düzenlenmesinin; malzeme taşıma maliyetini azaltmak, tesisteki boş alanların etkin kullanımı, bölümlerin şekillerine göre etkin kullanımı, tesisin genişletilebilmesinde kolaylık sağlaması, esneklik, ısıtma/soğutma gereksinimlerinin

karşılanması gibi bir çok amacı olmasına karşın genellikle malzeme taşıma maliyetlerini minimum düzeye indirmek amacıyla bu probleme yeni çözümler geliştirilir.

İmalattaki toplam işleyiş maliyetinin %20 - %50 oranını malzeme taşıma maliyetleri oluşturmaktadır ve etkin bir yerleşim bu maliyetleri en azından %10 -%30 oranında azaltabilir (Tompkins ve White, 1996).

Tesis yerleşimi düzenlenmesi (üretim), ya yeni bir üretim sistemi kurulduğunda ya da akış hacimlerinde veya güzergahlarında, üretim sistemine yeni kaynaklar (robotlar, otomatik güdümlü araçlar, vb.) eklendiğinde önemli değişiklikler meydana geldiğinde gerçekleştirilir (Dolgui ve Proth, 2010). Bazen mevcut tesislerde değişiklik yapmak çok masraflı ve hatta imkansızdır (çimento, petrol rafinerisi, kağıt, şeker vb.). Bu gibi endüstrilerde tesis yerleşimi düzenlenmesi tesis kurulumundan önce dikkatle planlanmalıdır. Özellikle çeşitli ürün üreten seri imalat endüstrilerinde ise, kapasite artışı ve yeni mamul eklenmesi gibi nedenlerle zamanla yeniden düzenlenme ihtiyacı doğacağından planlanan ilk tesis yerleşimi düzenlenmesinde çeşitli değişimler olabilir.

2.1.2. Tesis Yerleşimi Düzenlenmesinin Amacı

Tesis yerleşimi düzenlenmesinin en temel amacı, öngörülen kapasite ve kalite gereksinimlerini 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 (Erkut ve Baskak, 2003).

Panneerselvam (2006)'a göre göre tesis içi yerleşim düzenlemesinin amaçları şu şekilde sıralanmıştır:

- Üretim işlemlerini kolaylaştırmak
- Malzeme taşımayı ve maliyetini azaltmak
- İşçiler için kolaylık, güvenlik ve rahatlık sağlamak
- Donanım yatırımını azaltmak
- Üretim zamanının tümünü azaltmak
- İşlemlerin ve düzenlemelerin esnekliğini sağlamak
- Örgütsel yapıyı kolaylaştırmak

- Mevcut alanları daha verimli kullanmak

Kumar ve Suresh (2009) bunlara ek olarak şu amaçları da belirtmiştir:

- Tesis içindeki malzeme akışını düzenlemek
- İşlemdaki envanterin yüksek oranda devrini sağlamak
- İşçi, donanım ve mekanı verimli kullanmak
- Kübik alanların kullanımını verimli hale getirmek
- Üretim işlemlerinin ve düzenlemelerinin esnekliği

Gerçek bir üretim hattında tüm bu amaçların aynı anda gerçekleştirilmesi zor olmaktadır. Sürece göre yerleşim düzenlenmesinde taşıma maliyetlerini en aza indirme amacı öncelik taşırken ürüne göre yerleşim düzenlenmesinde üretim hattının dengelenmesi amacı öncelik kazanır. Tesisin alanı, ekipmanlar, malzeme ve araç-gereçler, işgücü gibi üretim girdilerinin etkin kullanımı üretim esnekliğine de kısıtlamalar getirmektedir, önemli olan tüm bu amaçların verimli bir şekilde dengelendiği optimum bir çözüm elde edebilmektir.

Tesis yerleşimi düzenlenmesinin kapsamı; bir üretim sisteminde tüm çalışma, malzeme ve depolama için gerekli alanların, makine ve teçhizatların doğru alanlara yerleştirilmesinin, ayrıca ürün, malzeme, araç-gereç ve işçilerin hareket edebilmeleri için yeterli alanların, bakım ve onarımın yapılabileceği atölyelerin, sağlık, beslenme ve eğitim için gerekli tesislerin aralarında düzen sağlanacak şekilde yerleştirilmesi ve düzenlenmesidir.

Quarterman vd. (1997)'ne göre tesis planlama (arazi, bina, ekipman, mobilya vb.) değer katmak için fiziksel kapasite sağlar. Uygun şekilde tasarlanmış bir tesis rekabet avantajı için önemli bir kaynaktır :

- düşük maliyetle çalışma
- hızlı teslim sağlama
- yeni ürünlere uyum sağlama
- çok sayıda çeşitli ürün üretme
- çok veya az miktarda ürün üretme
- en yüksek kalitede ürün üretme

- benzersiz hizmet veya özellik sağlama

Uygun şekilde yapılmış bir tesis yerleşimi düzenlenmesi, kendisini parçalara ayırmaya ya da yenilenmeye isteksiz olan bir organizasyonun değişim ihtiyacını gösterebilir:

- ürün odaklı iş istasyonu
- tesis içinde tesis odaklı fabrikalar
- azaltılmış stok ve dağıtım alanları
- kanban stok noktaları (merkezleri)
- hammaddelerin doğrudan dağıtımı
- bütünleşmiş destek alanlar

2.1.3. Tesis Yerleşimi Düzenlenmesinin Etmenleri

Bir tesisin veya işyerinin düzenlenmesinde etkili olan çok sayıda etmen vardır. Bu etmenleri sekiz ayrı grupta toplamak olasıdır (Erkut ve Baskak, 2003):

Malzeme: Tasarım, değişiklik, miktar, gerekli işlemler ve bunların sırası.

Makine: Üretim araçları, takımlar ve bunların kullanımı.

İnsan: Gözetim, denetim ve hizmet.

Hareket: Bölüm içi ve bölümler arası taşıma, değişik işlemler, depolamalar ve muayeneler.

Bekleme: Sürekli ve geçici depolamalar ve gecikmeler.

Hizmet: Bakım, muayene vb. hizmetler.

Bina: Binanın iç ve dış özellikleri, kullanım özellikleri ve donanım.

Değişim: Genişleme, esneklik.

2.1.4. Tesis Yerleşimi Düzenlenmesi Probleminin Gereksinim Duyduğu Veriler

İyi bir tesis yerleşimi düzenlenmesi, malzeme ve insan hareketinin maliyetini azaltırken bu maliyeti hesaplamak için bazı verilere gereksinim duymaktadır (Heragu, 2006):

- Departmanlar arası malzeme akışı veya hareketlerin sıklığı
- Departmanların şekli ve alan bilgisi
- Departmanlar için alan gereksinimleri
- Departmanlar için yerleşim kısıtları
- Departman çiftleri arasındaki yakınlık gereksinimleri

Bu verilerin hepsi gerekli değildir. Ancak, departmanlar arası malzeme ve insan hareketinin sıklığının, departmanların şekillerinin ve alanlarının bilinmesi başlangıç yerleşimini hazırlamak için gereklidir. Departmanlar arası hareket sıklığı departmanlar arası ilişkiyi belirlemek için kullanılır. Eğer bu sıklığı belirten veriler olmazsa tesis tasarımcısı departmanlar arası akış ve hareketler hakkında bir tahmin yapmak zorunda kalır. Tasarımcı her bir departmanın boyutlarını ve alan gereksinimini de bilmelidir. Tesis bir makine ya da iş istasyonu ise alan gereksinimine, çevresinde işçinin çalışabilmesi için gerekli alanın ve malzemenin taşınacağı koridor alanının da eklenmesi gerekmektedir.

2.1.5. Tesis Yerleşimi Düzenlenmesinin İlkeleri

Kumar ve Suresh (2009)'e göre tesis yerleşimi düzenlenmesinin ilkeleri şu şekilde sıralanmıştır:

- Birleştirme/bütünleşme ilkesi; etkin bir yerleşim düzeni, kaynakların optimum kullanımı ve maksimum verimlilik elde etmek için işçi, malzeme, makine ve destek hizmetlerin bütünleşik kullanımını sağlar.
- Minimum uzaklık ilkesi; bu ilke malzemelerin ve işçilerin minimum hareketiyle ilgilidir. Malzeme taşınmasının ve işçilerin hareketinin olabildiğince yakına ve doğrusal hareketler şeklinde olması tercih edilmelidir.
- Kübik alanların kullanımı ilkesi; etkin bir yerleşim düzeni, hem yatay hem dikey alanların kullanımını sağlar. Yalnızca zemin alanının optimum kullanımı yeterli değildir, yüksekliğin de verimli kullanılması gerekir.
- Akış ilkesi; etkin bir yerleşim, malzemeleri tamamlanma aşamasına doğru ileri yönde hareketini sağlar, geriye doğru hareket olmaz.
- Maksimum esneklik ilkesi; etkin bir yerleşim düzeni, fazla zaman ve maliyet gerekmeden değiştirilebilir, şimdiki yerleşim tasarlanırken ilerdeki gereksinimler hesaplanabilir.
- Güvenlik, gizlilik ve memnuniyet ilkesi; etkin bir yerleşim, işçilerin güvenliğini, memnuniyetini ve yangın, hırsızlık gibi durumlara karşı tesis ve makinelerin güvenliğini dikkate alır.
- Minimum taşıma ilkesi; etkin bir yerleşim, malzeme taşınmasını minimum düzeye kadar azaltır.

2.1.6. Genel Akış Modelleri

Üretim yapan bir tesisi tasarlamamanın ilk adımı; malzemeler, parçalar ve iş sürecindeki stoklar yoluyla oluşan genel akış modellerini belirlemektir. Akış modelleri, başlangıçtan son ürüne yani hammaddenin işlenmeye başlamasından yarı mamül olmasına, yarı mamulden de son ürün olmasına kadar geçen tüm akış sürecini içerir (Heragu, 1997).

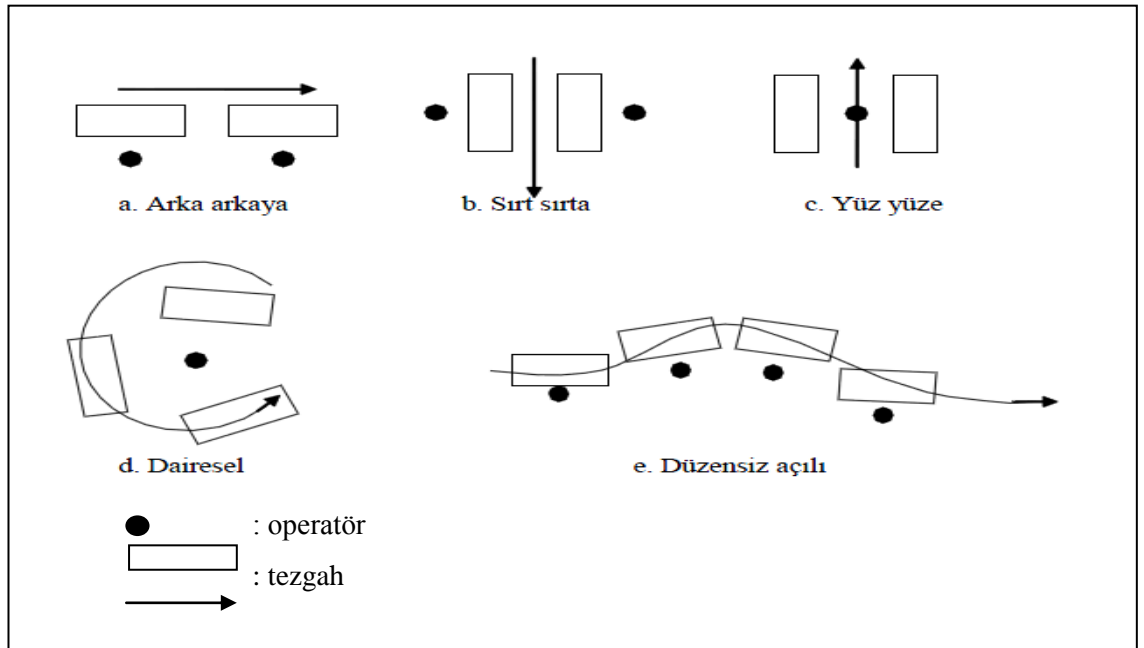
Akış modellerini, “iş istasyonları içindeki akış”, “bölümler içindeki akış” ve “bölümler arasındaki akış” olarak ayrı ayrı açıklamak daha doğru olacaktır (Tompkins vd., 1996):

2.1.6.1. İş istasyonları İçindeki Akış

Bu akış oluşturulurken, dikkat edilmesi gereken işçilerin hareketleri ve ergonomidir. İş istasyonları içerisindeki akış, simetrik, eş zamanlı, doğal, ritmik ve alışılmış olmalıdır.

2.1.6.2. Bölümler İçindeki Akış

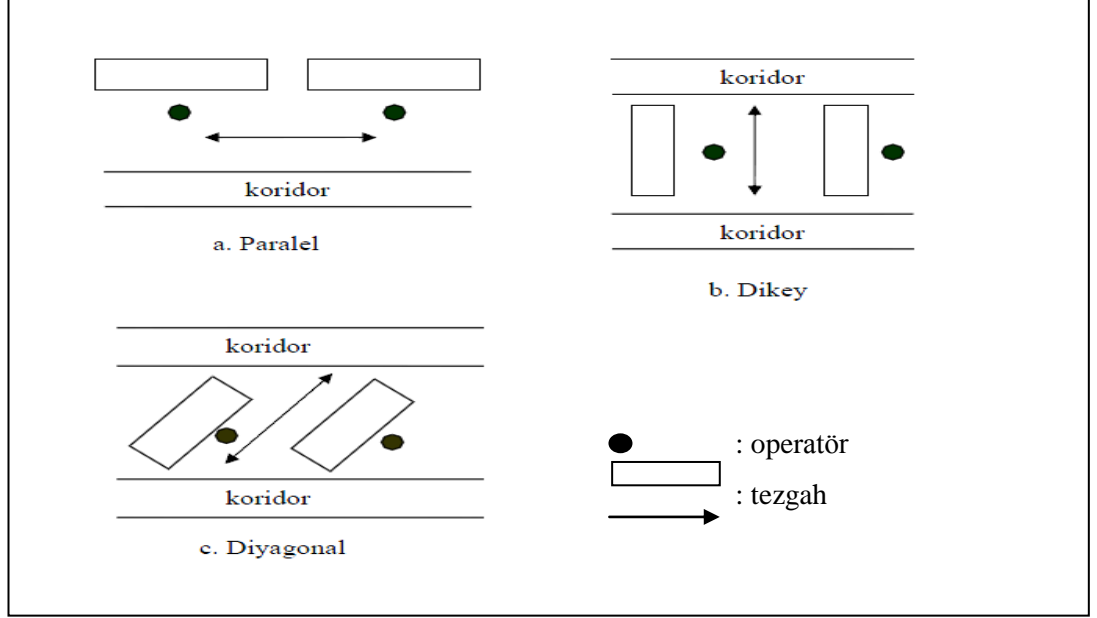
Bölümler içerisindeki akış, bölümün tipine bağlıdır. Bir üretim bölümünde, iş akışı ürün akışını izler. Ürün akışı tipik olarak Şekil 2.1’de gösterilen modellerden biri gibidir.



Şekil 2.1 : Ürün Akışı Modelleri (Tompkins vd., 1996)

Bir süreç bölümünde, bölümler içerisindeki iş istasyonları arasındaki akış küçük olmalıdır. Akış tipik olarak iş istasyonları ve koridorlar arasında ortaya çıkar. Akış modelleri, iş istasyonlarının koridorlara göre yönlerine bağlı olarak oluşmuştur. Şekil 2.2’de iş istasyonu-koridor yerleşimine bağlı olarak oluşan üç tane akış modeli

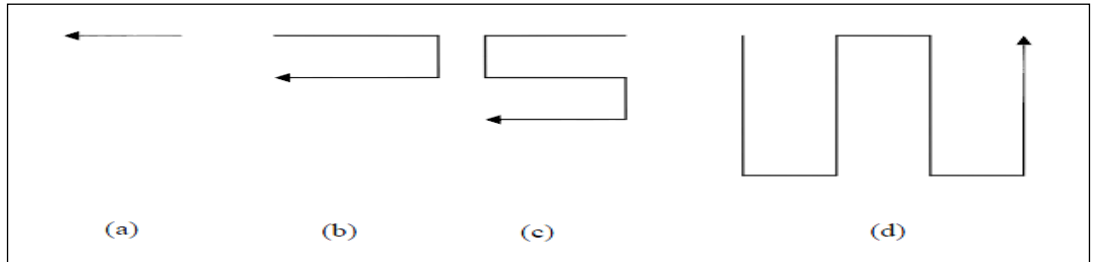
görülmektedir. Hangi iş istasyonu-koridor düzeninin seçileceği, iş istasyonlarının kapladığı alanlar, kullanılabilir alanlar ve taşınacak malzemenin miktarı/büyüklüğü arasındaki ilişkiye bağlı olarak değişmektedir.



Şekil 2.2 : İş İstasyonu-Koridor Yerleşimine Göre Akış Modelleri (Tompkins vd., 1996)

2.1.6.3. Bölümler Arasındaki Akış

Bölümler arasındaki akış, bir tesisteki tüm akışı değerlendirmek için sıklıkla kullanılan bir ölçüttür. Akış genellikle, Şekil 2.3’de gösterilen dört tip akış modelinin kombinasyonundan oluşmaktadır. Şekilde gösterilen akış modellerinin biraraya getirilmesindeki önemle dikkate alınması gereken, giriş ve çıkışın yeridir. Çizim planının veya bina yapısının bir sonucu olarak, giriş ve çıkış yerleri genellikle sabittir ve tesis içerisindeki akış, bu kısıtlar altında ortaya çıkmaktadır.



Şekil 2.3 : Genel Akış Modelleri: (a) I tipi akış, (b) U tipi akış, (c) S tipi akış, (d) W tipi akış (Tompkins vd., 1996)

2.1.7. İş Akış Tipleri

Erkut ve Baskak (2003) belirli bir ürün veya ürün grubunun üretimi için gerekli makine ve insan gücü gereksinmesinden sonra, bu makine ve işgören grubunun tesiste birleştirilerek yerleştirilmesi gerektiğini belirtmişlerdir. Bu düzenlenmede temel amaç; hammadde girişinden, bitmiş ürünün müşteriye teslim edilmesine kadar söz konusu olan süreçlerde en az ağırlık, uzaklık ve maliyetle taşınmayı sağlayacak bir iş akışının oluşturulmasıdır. Malzemelerin işletmedeki akışı, tesis düzenlenmesini saptayan önemli öğelerden biridir. İşyeri düzenlenmesinde ulaşılmak istenen amaçlardan biri de makinelerin değişmez olarak düzenlenmesi değil, malzemenin fabrikada en iyi akışının sağlanmasıdır.

Malzemenin etkin biçimde akışı; malzeme aktarma giderlerini, yarı ürünlerin miktarını, yarı ürünlere bağlanan ana mal ve alanı, toplam üretim süresinin uzunluğunu önemli ölçüde saptama zorunluluğunu ortaya koyar. Böylece tesis düzenlenmesinde etkin bir akışın yanı sıra en küçük hareketin de sağlanması gerekli olur. Tüm bu öğelerin her biri, akış sistemlerinin titizlikle incelenmesi için yeterli sayılır. İş akışı sistemi ayrıca, denetim ve kontrol mekanizmasını da etkiler. Yalın iş akışı hatları, kontrol amaçları için ideal olurlar ve bu durumda da gözetim gözle yapılabilir. Karışık iş akışı sistemi, genellikle karışık kontrol sistemini doğurur.

Sürece, ürüne ve programa ilişkin temel bilgiler toplandıktan sonra tesis düzeni tasarımcısı, malzemelerin, araç ve gereçlerin ve işgörenin akışının analizine geçer. Tesis düzenleme gerçekte ürünün, hammadde durumundan işlenmiş ürüne kadar olan aşamalardaki akışını kolaylaştırmak olduğundan, malzemelerin yani işin akışı incelenmelidir.

2.1.7.1. Akış Sistemini Etkileyen Öğeler

Bir tesis ister sürece, isterse ürüne göre düzenlemiş olsun iş akışı modellerinin özellikleri ayrıntılı olarak göz önünde bulundurulur. İş akışını etkileyen başlıca etmenler şu şekilde sıralanabilir (Erkut ve Baskak, 2003):

- 1) Dış ulaşım kolaylıkları,
- 2) Üründeki parçaların sayısı,

- 3) Her bir parçadaki işlemlerin sayısı,
- 4) Her bir parçadaki işlemlerin sırası,
- 5) Alt montajların sayısı,
- 6) Üretimi yapılacak ürün sayısı,
- 7) İş istasyonları arasındaki akış,
- 8) Elverişli alanın hacmi ve biçimi,
- 9) Süreçlerin/işlemlerin etkisi,
- 10) Akış tipleri,
- 11) Ürüne ve sürece göre tesis düzenlenmesi,
- 12) Hizmet bölümlerinin konumları,
- 13) Üretim bölümlerinin konumları,
- 14) Bölümlerin özel gereksinimleri,
- 15) Malzeme depolama,
- 16) İstenilen esneklik,
- 17) Bina.

2.1.7.2. İş Akış Modelleri

İş akış modelleri özelliklerine göre ikiye ayrılabilir (Erkut ve Baskak, 2003):

- 1) Üretim hattı bakımından iş akış modelleri:
 - a) Doğrusal (düz),
 - b) U-biçimli,
 - c) S-biçimli,
 - d) Bükümlü (helezoni).
- 2) Montaj hattı bakımından iş akış modelleri: Tarak biçimli,
 - a) Ağaç biçimli,
 - b) Dal,
 - c) Havai (üst üste).

Tüm bu iş akışı modelleri Şekil 2.4.'te görüldüğü gibidir.

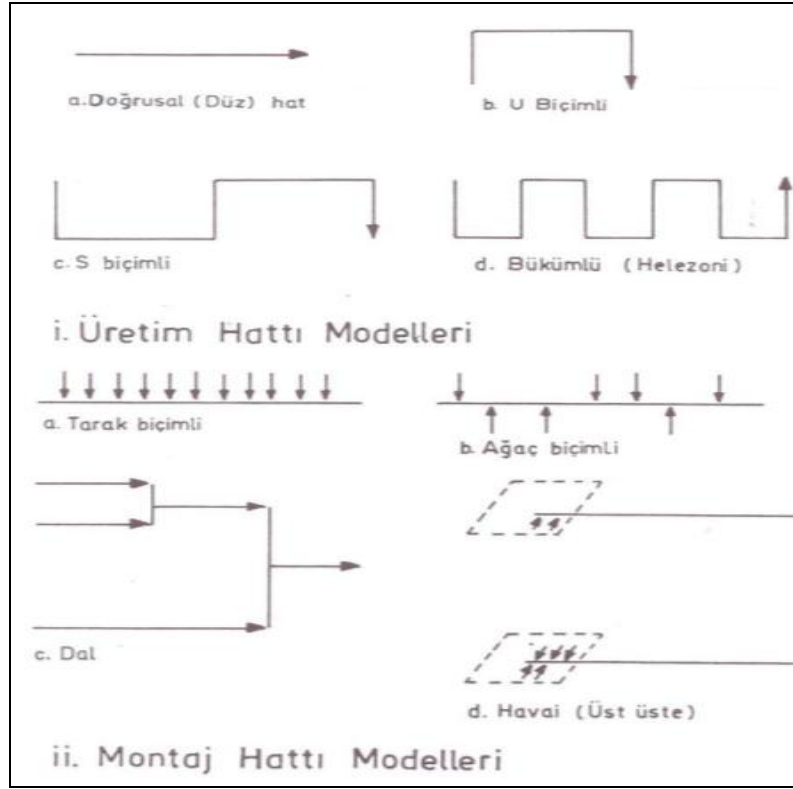
Tesis düzenlenmesindeki amaç, en az geri dönüşle, en kısa uzaklıkta üretimi gerçekleştirmektir. Yani toplam malzeme/ürün aktarma uzaklığını en alt düzeye indirmektir.

Doğrusal hat akış tipi, çok dar veya üretim hattının genişliği kadar bir binayı gerektirebilir. U-biçimli üretim hattı ise, malzeme veya ürünlerin gönderiminin aynı bina içerisinde yapılmasına, bu da işletme ve denetim noktalarının birleştirilmesine olanak sağlar. Üretim kimi durumlarda S ya da bükümlü hatlarda gerçekleştirilebilir.

Montaj hattı biçimlerinden tarak modelinde ana montaj hattı aynı yönden, ağaç modelinde ise her iki yönden gelen alt-montaj hatlarıyla beslenir. Ana montaja gelen aynı sayıdaki alt-montaj hatları açısından, tarak tipi modele oranla ağaç tipi montaj hattında montaj işlemleri daha kısa sürede tamamlanabilir. Tarak tipi montaj hattı daha çok uzun bir duvar veya koridor türündeki yerleşim için uygundur.

Ağaç türü montaj hattı ise ana montaj hattının üretim alanının merkezine yerleştirilmesi ve yanlardan gelen alt montaj hatlarıyla beslenmesi durumunda kullanılmaktadır.

Dal tipi modelde, alt- montaj hatları üretim hattı boyunca ilerlerken, zaman zaman diğer alt-montaj hatları ile birleşerek en sonunda ana montaj hattını oluştururlar. Bu biçimde son montaj hattına yaklaştıkça, alt-montaj hatlarının sayısı azalır ve sonuçta ana-montaj hattı kurulmuş olur. Tarak ve ağaç türü montaj hattı iş akış türlerine göre daha düzensizdirler. Havai (üst üste) iş akışında ise montaj hattı, yukarı mekandan ilerleyerek aşağıya inmekte ve alt mekanda tamamlanmaktadır.



Şekil 2.4 : İş Akış Modelleri (Erkut ve Baskak, 2003)

Montaj hattı iş akış türlerinden hangisi seçilirse seçilsin, alt montajlar arasında birbirini izlemesi gereken işlemlerin, olası olduğu kadar birbirine yakın olmasına çalışılır. Böylece taşıma sürelerinin en azda kalması sağlanarak ürün üretimi için harcanan süre de azalmış olmaktadır. Toplam üretim sistemindeki verimlilik ve maliyet açısından montaj hatları, var olan alandan, olası en yüksek yararı sağlayacak biçimde ve uzunlukta düzenlenmektedir.

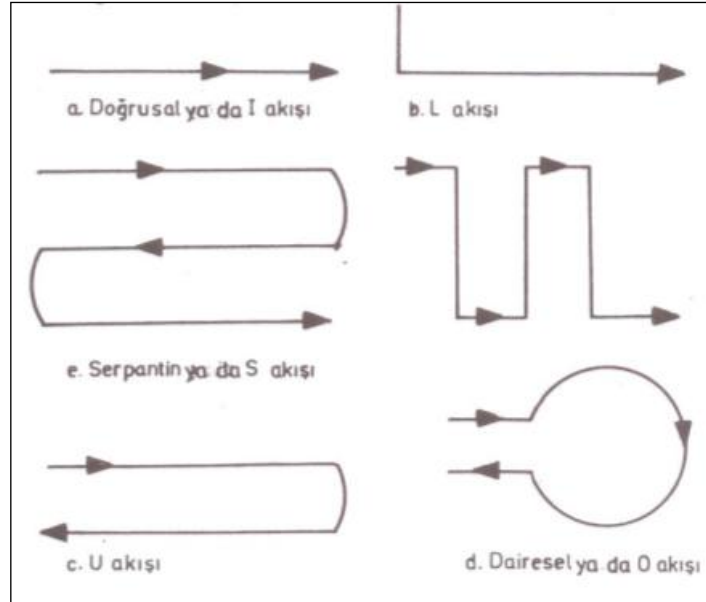
Bir başka düşünceye göre iş akış sistemleri ikiye ayrılır (Erkut ve Baskak, 2003):

1) Yatay iş akışı hatları:

- a) Doğrusal ya da I akış hattı,
- b) L akış hattı,
- c) U akış hattı,
- d) Dairesel ya da O akış hattı,
- e) Yılankavi ya da S akış hattı.

2) Dikey iş akışı hatları:

- a) Yukarıya veya aşağıya doğru süreç akış sistemi,
- b) Merkezi veya merkezi olmayan kaldırma sistemi,
- c) Tek yönlü veya geri dönüşlü akış,
- d) Dikey veya eğimli akış,
- e) Tek veya çoğul akış,
- f) Binalar arası akış.



Şekil 2.5 : Yatay İş Akışı Hatları (Erkut ve Baskak, 2003)

Yatay iş akışı hatları Şekil 2.5'te gösterilmiştir.

Yatay iş akışı hatlarından doğrusal ya da I akış hattı, en basit akış türüdür. Ancak gönderme ve kabul etme işlemleri için ayrı ekiplerin oluşturulması gerekmektedir.

L biçimi akış, doğrusal akışın uygulanmadığı veya oluşturulmasının yüksek maliyetler gerektirdiği durumlarda kullanılmaktadır.

U biçimi akış, yönetimi basit olması yanında, gönderme ve kabul etme işlemlerinin aynı ekip tarafından yapılabilmesi nedeniyle oldukça beğenilmektedir.

Dairesel akış, bitiş noktasının başlangıç noktasına çok yakın oluşturulması istendiği takdirde uygulama alanı bulmaktadır.

Yılkavi akış, üretim hattının çok uzun olması nedeniyle üretim alanında zigzag oluşturulması gerektiği zaman kullanılmaktadır.

2.1.8. Tesis Yerleşim Tipleri

Üretimde kullanılacak iş akış tiplerine karar verildikten sonra tesislerin, departmanların ve makinelerin konumları belirlenir. Bunun için genellikle deneme yanılma yöntemi kullanılmaktadır. Pek çok mümkün çözüm olabilir ama problemin kendi özelliklerine göre amaçlara en uygun yerleştirme düzeni bulunmaya çalışılır. Değerlendirilen amaçlara ve kısıtlara göre, bazıları üzerinde çeşitli hesaplama yöntemlerinden yararlanarak seçim yapmak mümkündür. Kumar ve Suresh (2009) yerleşim düzenini 5 gruba ayırmıştır:

1. Sürece göre yerleşim
2. Ürüne göre yerleşim
3. Sabit pozisyonlu ürüne göre yerleşim
4. Hücresel yerleşim
5. Birleşik yerleşim

Sürece Göre Yerleşim

Sürece göre yerleşim toplu üretimi için tavsiye edilir. Sürece göre yerleşimde bir alanda gruplanmış bütün makineler benzer işlemleri yapar. Torna, freze vb. benzer gruplarda sınıflanır. Böylece, sürece göre yerleşimde tesislerin düzeni işlevlerine göre beraber gruplanır. Malzeme akış hattı bir işlem alanından diğerine doğru üründen ürüne değişime uğrar. Genellikle hat uzundur ve geriye doğru akış mümkündür. Sürece göre yerleşim normalde üretim miktarı ürüne göre yerleşim için yeterli olmadığı zaman kullanılır. Genellikle iş atölyeleri üretilen ürünlerin çeşitliliği ve düşük üretim miktarları nedeniyle sürece göre yerleşim kullanır.

Yararları:

- Makineler daha iyi kullanılmakta ve daha az makine gerekmektedir.
- İşgücü ve donanım esnekliği mümkündür.
- Daha az makine yatırımı gerekir ve genel amaçlı makinelerin maliyeti daha az olur.
- Üretim tesislerinin kullanımını artar.
- Makinelerin ve işçilerin iş dağılımında yüksek derecede esneklik sağlanır.
- Görevlerin farklılığı ve işlerin çeşitliliği işi ilgi çekici ve ilginç yapar.
- Yöneticiler kendi departmanlarındaki işler ile ilgili çok bilgili olur.

Sakıncaları:

- Bu durumda malzeme taşıma işleminde geriye doğru akışlar ve uzun mesafeli hareketler olabilir ve malzeme taşıma maliyeti yükselir.
- Malzeme taşıma mekanikleşemez.
- Geriye doğru akış yapan envanterler arttıkça ve işlenmemiş ürün stoğu arttıkça üretim süresi uzar.
- Kurulum sayısına bağlı olarak verimlilik düşer.
- Çıktı zamanı (işleme giriş ve çıkış arasındaki zaman aralığı) uzar.
- Kazanç ve alan yarı mamul stoğuna göre birbirine bağlıdır.

Ürüne Göre Yerleşim

Bu yerleşim düzeninde makineler ve yardımcı hizmetler ürünün işlem sırasına göre yerleştirilir. Bir ya da daha fazla ürünün üretim miktarı fazlaysa birim maliyeti azaltmak ve malzeme akışını verimli hale getirmek için tesisler düzenlenebilir. Özel amaçlı makineler gereken hızlı ve güvenilir işleri gerçekleştirmek için kullanılırlar. Ürüne göre yerleşim, bir ürünün üretim miktarı üretiminin yapılması için ayrı bir üretim hattını gerektirdiği zaman kullanılır. Tam bir ürüne göre yerleşim düzeninde makineler farklı ürünler ile paylaşılabilir. Bu nedenle üretim miktarı donanımın verimli kullanımını sağlamak için yeterli olmalıdır.

Yararları:

- Üretimin akışı düzgün ve mantıklı olmalıdır.
- Yarı mamul stoğu küçüktür.
- Toplam üretim süresi kısadır.
- Malzeme taşıma maliyeti minimumdur.
- Üretim, planlama ve kontrol sistemlerini basitleştirmek mümkündür.
- Geçici depolama ve iş geçişleri ile daha az alan meşgul olur.
- Düzgün akış ve mekanik taşıma sistemleri ile malzeme taşıma maliyeti azalır.
- İyi bir hat dengeleme atıl kapasite ve darboğazları ortadan kaldırır.
- Üretim döngüsü malzemelerin kesintisiz akışından dolayı kısadır.
- İşlenmemiş ürün stoğu azdır.
- Kalifiye olmayan işçiler işi öğrenip işi yapabilirler.

Sakıncaları:

- Üretim hattındaki bir makinenin arızalanması akış hattında sonra gelen makinelerin durmasına neden olabilir.
- Ürün tasarımındaki bir değişiklik yerleşim düzeninde büyük değişiklikleri gerektirebilir.
- Üretim hızını darboğazdaki makine belirler.
- Donanım için yüksek yatırım gereklidir.
- Esnek değildir. Üründeki bir değişiklik tesisin yeniden düzenlenmesini gerektirir.

Sabit Pozisyonlu Ürüne Göre Yerleşim

Bu yerleşim tipi proje tipi yerleşim olarak da adlandırılır. Bu yerleşimde, malzeme, veya temel elemanlar sabit bir yerleşimde dururken araçlar, makineler, işçiler ve diğer malzemeler bu alana taşınır. Bu yerleşim tipi bir veya bir kaç ağır ürünün üretilmesinde ve montajı çok sayıda taşıma maliyeti yüksek ağır parça içeriyorsa uygundur. Bu yerleşim tipinin en büyük avantajları şunlardır:

- İş yayılımına ve işçilerin yeteneklerini arttırmaya yardımcı olur.
- İşçiler işi yaparken ilgi duydukları bir ürünle kendilerini tanımlarlar.
- Bu yerleşim tipi çok büyük oranda esnektir.
- Yerleşim düzenlemesi için sermaye yatırımı daha azdır.

Hücreyel Üretime Göre Yerleşim

Grup Teknolojisi, parçaları analiz edip karşılaştırarak onları benzer özellikteki aile gruplarına ayırır. GT sadece sürece göre yerleşim ile sadece ürün akışına göre yerleşimin bir karışımını geliştirebilir. Bu teknik işletmeler için küçük partiler halinde çeşitli ürünlerin üretimine uygunken akış hattının ekonomikliği ile avantajlarına sahip olduğu için çok faydalıdır. GT uygulaması iki temel adım içerir; ilk adım parça ailelerini veya gruplarını belirlemek, ikinci adım ise belirli bir bileşen ailesini işlemek için kullanılan donanım alanını düzenlemektir. Tesis içinde küçük tesisler şeklinde görünür. GT üretim planlama süresini, işlerini ve kurulum süresini azaltır. Böylece hücreyel üretime göre yerleşim (grup yerleşimi) ürüne göre yerleşimin ve sürece göre

yerleşimin birleştirilmesidir. Bu yerleşim her iki yerleşimin de avantajlarını sağlar. Hücresel üretimde asıl amaç toplam taşıma ve donanım maliyetlerini minimum yapmaktır. Bu sebeple çok amaçlı yerleşim düzeni olarak adlandırılır.

Hücresel üretime göre yerleşim ;

- bileşenin standardizasyonunu ve rasyonelizasyonunu
- tahminlerin güvenilirliğini
- etkin makine çalışması ve verimliliğini
- müşteri hizmetlerini

arttırabilirken;

- kağıt işlerini ve genel üretim zamanını
- ilerlemeyen işi ve iş hareketlerini
- genel maliyeti

azaltabilmektedir.

Birleşik Yerleşim ;

Ürüne göre yerleşimin ve sürece göre yerleşimin birleştirilmesi her iki tipin de avantajlarını içerir. Birleşik yerleşimde bir parça farklı tiplerde ve boyutlarda üretilebilmesi mümkündür. Makineler sürece göre yerleşimdeki gibi düzenlenir fakat farklı tip ve boyutta ürünler üretmek için bir sıraya göre süreç gruplanır. Şu belirtilmelidir ki işlem sırası çeşitli ürünlerde ve boyutlarda aynı kalır.

2.1.9. Tesis Yerleşimi Düzenlenmesi Problemlerinin Sınıflandırılması

Tesis yerleşim sorunları; ürün, hizmet; miktar, tasarım ya da prosesinde değişiklik yapılması, yeni bir makine-teçhizatın devreye alınması gibi durumlarda ortaya çıkabildiği gibi yeni bir tesise taşınması veya yeni bir işletmenin kurulması şeklinde de ortaya çıkabilmektedir. Uzun zaman boyunca biriken problemlerin çözümü için de tesis yerleşim düzeninin yeniden gözden geçirilmesi gereği duyulabilir. Malzeme aktarma zamanlarının toplam üretim zamanına göre yüksek olması, kalabalık çalışma ortamları, yetersiz işlemler, makinelerde darboğazlar, açıklanamayan gecikmeler ve boş zamanlar ve kötü çalışma ortamı böyle durumlara örnek olarak gösterilebilir.

Sebepleri dikkate almaksızın temel olarak tesis yerleşim problemleri şu şekilde sınıflandırılabilir (Erkut ve Baskak, 1996):

- Var olan düzenlenmede küçük değişikliklerin yapılması,
- Var olan düzenlenmenin yeniden ele alınması,
- Var olan tesislere taşınma,
- Yeni bir işletmenin,/fabrikanın kurulması

2.2. TESİS YERLEŞİMİ DÜZENLENMESİ İÇİN GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİ

Tesis yerleşimi düzenlenmesi verilen yerleşim yerinde belirli bir alana gereksinim duyan bir grup departmanın yerleştirileceği alanın belirlenmesi ile ilgilidir. Departmanların değerlendirilmesi, yakınlık matrisi ile gösterilebilecek departmanlar arası gezilerin (gidiş-geliş) sayısına dayanır. Departmanlar arası gezi sayısı bir departmanın bir diğerinin yanına yerleştirilmesinin karar verilmesinde kullanılabilir. Verilen bölüm düzeninde problem, bölümler arası yakınlık ilişkileri ve mesafelere dayanan bir optimizasyon fonksiyonu sağlayan bir yerleşim düzeni bulmaktır (Bhowmik, 2008)

2.2.1. Geleneksel Yaklaşımlar

Pek çok literatür kaynağı tesis yerleşim yöntemlerini yapısal açıdan algoritmik ve prosedürel olmak üzere iki büyük kategoriye ayırırlar. Yani tesis düzenlenme problemi hem tasarım hem de optimizasyon problemidir. Algoritmik yaklaşımlar; genellikle çözümü elde edilebilen amaç fonksiyonunun yerini tutan bir fonksiyona ulaşmak için tasarım kısıtlarını ve amaçlarını basitleştirirler. Mevcut literatürün çoğunluğu algoritmik yaklaşımları anlatır. Tesis yerleşimi düzenlenmesi çok amaçlı yapısı ve bilgi gereksinimi nedeni ile zaman ve emek gerektirir. Günümüze kadar yapılan bir çok araştırmada bu gereksinimlerin karşılanması için çözüm yöntemleri geliştirilmeye çalışılmıştır. Algoritmik yaklaşımların asıl amacı malzeme taşıma maliyetlerini minimum kılmak için akış uzaklıklarını minimum düzeye indirmeye çalışmaktır. Prosedürel yaklaşımlar ise uzmanların tecrübelerine dayanır (Heragu,1997).

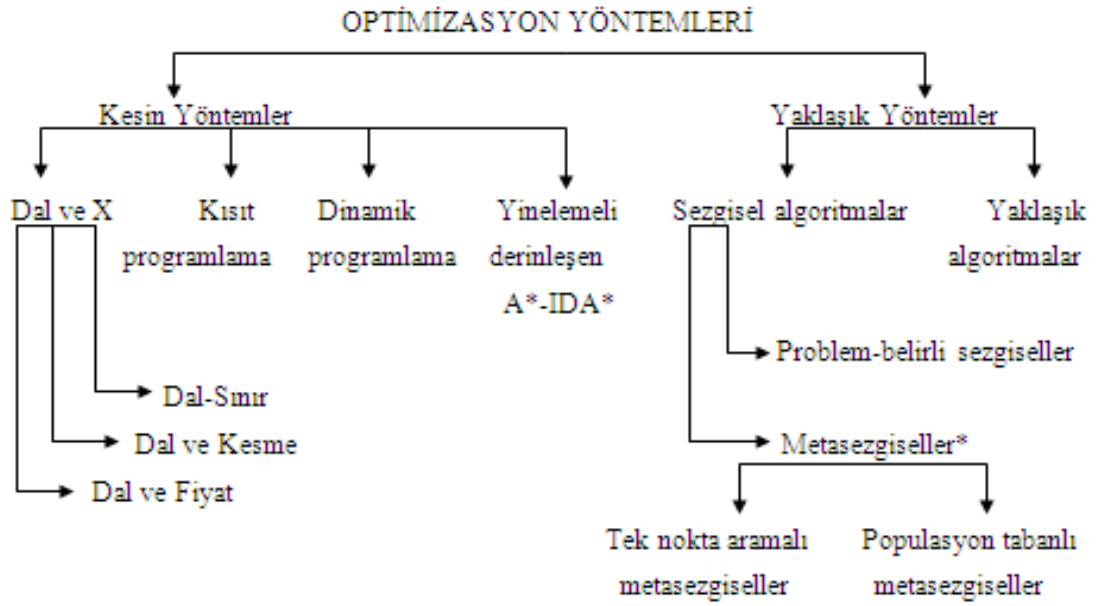
Algoritma, mekanik davranan kişiye veya bir makineye, bir takım verilerden yola çıkarak ve sonlu sayıda aşamalardan geçerek, belli bir problemi çözme imkanı veren, çok kesin komutlar bütününden oluşmaktadır. Bir algoritmanın çalışmasındaki mutlak zorunluluk, her türlü belirsizlikten arınmış olmasıdır. Bir algoritmanın yürütülmesi, her biri bir komutla belirlenen bir etkiler dizisi oluşturur ve bu dizi önceki komutun yürütülmesinin sona ermesiyle birlikte yürütülmeye başlar. Genel olarak algoritmaların aşağıdaki özellikleri ele alınmaktadır (Nabiyev, 2003):

- Genellik
- Kesin sıralılık
- Sırayı belirleyen kumanda yapısı
- Sonluluk ve neticelik

Tesis düzenleme probleminin aynı zamanda bir optimizasyon problemi olduğu ifadesine dayanarak optimizasyon tanımı kısaca şu şekilde verilebilir: Optimizasyon mevcut koşullar altında en iyiyi arama sürecidir. Daha teknik bir ifadeyle, belli kısıtlar altında (eğer varsa) bir fonksiyonun maxima ya da minimasını bulmayla ilgilenen bir matematiksel disiplindir. Optimizasyon modelleri ise bir sistem çıktısını en iyilemek için, sistemin ilişkilerinin matematiksel ifadelerle tanımlanmış biçimidir.

Talbi (2009) klasik optimizasyon yöntemlerini Şekil 2.6'da gösterildiği gibi gruplamıştır.

Optimizasyon problemlerini çözmek için, çoğu sadece belli tipteki problemler için uygun bir çok algoritma bulunmaktadır. Uygun bir çözüm metodu bulmak için problemi iyi tanımak gerekir.



Şekil 2.6 : Klasik Optimizasyon Yöntemleri (Talbi, 2009)

Kesin metotlar her zaman optimal çözümler elde ederler ve bu optimallığı garanti ederler. Kesin metodlar problemin tüm çözüm uzayında arama yaparak tüm aday çözümleri hesaplayan ve en iyi sonucu bulan yöntemlerdir. Kesin metodlar kesin bir sonuç bulurlar; problem boyutu büyüdükçe çözümün bulunması çok uzun zaman harcamaktadırlar. Kesin metotlar sınıfında dinamik programlama, yöneylem çalışmalarıyla geliştirilmiş dal-X (dal-sınır, dal-kesme, dal-fiyat) algoritmaları, kısıt programlama ve yapay zeka çalışmalarında geliştirilmiş A* (A*,IDA*-yinelemeli derinleşen) algoritmaları bulabiliriz. Kesin metotlar zor problemlerin küçük boyutlu örneklerinde kullanılabilir. Bazı kesin metotlar geniş iş istasyonu ağlarının geliştirilmesinde kullanılabilir (Talbi, 2009).

Kısıt programlama, ağaç arama kavramları ve mantıksal sonuçlar çerçevesinde inşa edilmiş bir dildir. Kısıt programlamada optimizasyon problemleri bir takım kısıtlar tarafından bağlanmış bir grup değişken ile modellenir. Değişkenler tam sayıların sonlu alanında kendi değerlerini alabilirler. Kısıtlar matematiksel ya da sembolik şekillere sahip olabilirler (Talbi, 2009).

Dinamik programlama, Optimizasyon stratejisinin bir problemi daha küçük alt problemlere bölmeye dayalı olduğu durumlarda kullanılır. Bu alt problemleri ilişkilendiren denkleme Bellman Denklemi denir.

Halaç (2001), dinamik programlamayı şu şekilde tanımlamıştır; çözümü güç olabilecek bir problemi daha küçük alt problemlere ayrıştırarak çözüme ulaşmayı sağlayan bir optimizasyon tekniğidir. Bu nedenle dinamik programlamanın matematik modelinin birbirleriyle bağlantılı alt problemlere ayrılabilme özelliğinde olması gerekir. Bu bağlamda dinamik programlama, birbiri ile ilişkili kararlar serisini çözüme kullanılan sayısal bir tekniktir. Bu teknikte, mevcut sistem birbiri ardından işlem gören bileşenlere ayrılmakta ve ardışık iki işlem arasında fonksiyonel bir bağıntı kurulması yoluna gidilmektedir. Dönüşüm fonksiyonu (yineleme denklemi) olarak adlandırılan bu bağıntılar yardımıyla, bir önceki kararın içerdiği bilgilerden bir sonraki adımın çözümünde yararlanma yoluyla, optimizasyon adım adım gerçekleştirilmektedir. Her adımda bulunan çözüm, kendi başına problemin çözümü olmayıp, sadece optimal çözümün bir parçasını belirleyen bilgiyi içermektedir.

Tesis yerleşimi düzenlenmesini bir optimizasyon ve tasarım problemi olarak ele alan başlıca yaklaşımlar alt başlıklarda ele alınmıştır.

2.2.1.1. Tasarım Problemleri

Tasarım problemi tesis yerleşim düzenlenmesi problemini sistematik tesis düzenlenme problemi ve mühendislik tasarım problemi olarak ele alır.

i) Sistematik tesis düzenlenme problemi (STD): Uzun yıllardır kullanılmasının birincil sebebi tesis tasarımı için basit ve adım adım bir yaklaşım olmasıdır. STD tekniği temelde 4 adımdan oluşmaktadır (Heragu, 2006):

Adım 1: Departmanların Yerleştirileceği Yerleşim Yerine Karar Verme: Bu adım, departmanlar için yerleşim yerinin tanımlanmasını içermektedir. Örneğin, bu alan binanın kuzey tarafında, batı tarafında ya da eski binaya bitişik başka bir bina içerisinde olabilir.

Adım 2: Genel Yerleşimin Kurulması: Bu evrede, bölümler arasındaki akış, yakınlık durumları, her bir bölüm için alan gereksinimleri eldeki kullanılabilir alanla ve bütçe gibi kısıtlarla ilişkisi düşünülerek belirlenir ve yerleşim tipine karar verilir. Oluşturulan plânlar, maliyet ve maliyet dışı etmenler temel alınarak geliştirilir ve bölümler ve genel çalışma alanları için bir yerleşim seçilir.

Adım 3: Ayrıntılı Yerleşim Planlarının Oluşturulması: Adım 2’de; belirlenmiş olan departmanların gerçek pozisyonları, departmanların yerleşim düzenleri ve her bir makinenin, yardımcı donanımın, dinlenme ve temizlik odaları ve muayene istasyonu gibi destek hizmetlerin yerleri ve yerleşimleri ile ilgili ayrıntı sağlanmaz. Adım 2, departmanların yerleşim sorununu ele alırken, Adım 3, her bir departman içerisindeki yerleşimi ve yerleşim detaylarını ele alır.

Adım 4: Seçilen Yerleşimin Kurulması: Ayrıntılı yerleşim düzeni, etkilenen çalışanlar, idarî elemanlar ve yöneticiler gibi ilgili kişiler tarafından onaylandıktan sonra uygun yerleşim düzeni seçilir. Bu planlar ve krokiler bütün detayları göstermelidir, çünkü bu planlar yeni tesislerin taşınmasında veya kurulumunda kullanılır. 4. Adımda tesislerin taşınması/kurulumu ve makinelerin taşınması için sermaye ve zaman tahsis edilir.

Yukarıdaki adımlar içerisinde en önemli olanları Adım 2 ve 3’dür. STD için gereken girdi verileri beş sınıfta toplanabilir:

Ürün (P): Üretilecek ürün tipleri

Miktar (Q): Her bir parça tipinin üretim hacmi

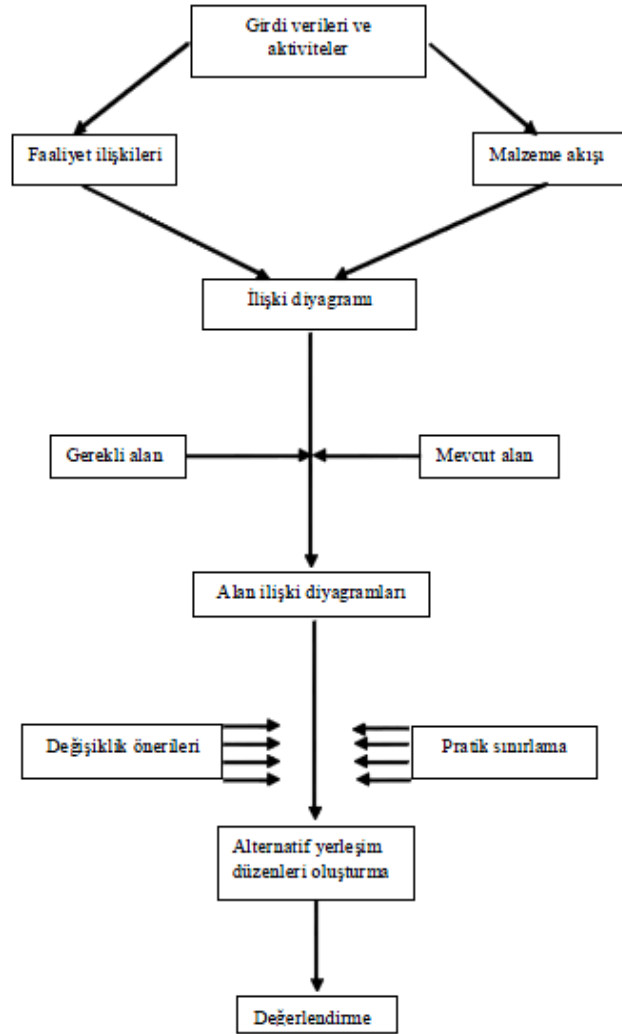
Rotalama (R): Her bir parça tipi için işlem sırası

Hizmetler (S): Destek hizmetler, dolaplar, muayene istasyonu vb.

Zamanlama (T): Parça tipleri ne zaman üretilecek ve üretim sırasında hangi makineler kullanılacaktır.

P-Q-R verisi ile, bir nereden nereye malzeme akış matrisi oluşturulur. Bu matris her bir makine çifti arasındaki akış yoğunluğunu göstermektedir. Benzer olarak, P-Q-S verisi ile ilişki şeması oluşturulur. Daha sonra, akış matrisi ve ilişki şemasından yararlanarak ilişki diyagramı oluşturulur. Bir sonraki adım, alan gereksinimlerinin ve yeterliliklerinin belirlenmesidir. Bu iki bilgiden ve ilişki diyagramından yararlanarak alan-ilişki diyagramı oluşturulur. Diğer etmenler (malzeme elleçleme yöntemleri, depo donanımı

vb.) ve kısıtlamalar (maliyet, var olan yapılar veya kirişler, personel ve donanım güvenliği, enerji yeterliliği) de hesaba katıldıktan sonra, alan-ilişki diyagramı, seçenek yerleşim düzenleri oluşturmak için değiştirilir. Bu aşamada, genellikle iki veya üç tane seçenek yerleşim oluşturulur, daha fazlası zaman alır ve önemli ölçüde yeni bir bilgi sağlamaz. Son olarak, her bir seçenek yerleşim düzeni, maliyet ve diğer fiziksel olmayan etmenlere bağlı olarak değerlendirilir ve aralarından en uygun olanı seçilir.



Şekil 2.7 : Sistematik tesis düzenlenme yordamı (Francis ve White, 1974)

ii) Mühendislik tasarım problemi: Aşağıda belirtilen altı adımdan oluşmaktadır (Tompkins ve White, 1984):

1. Problemi tanımla
2. Problemi analiz et
3. Seçenek tasarımlar oluştur
4. Seçenekleri değerlendir
5. Yeğlenen tasarımı seç
6. Seçilen tasarımı oluştur

Mühendislik tasarım süreci, tesis planlamaya uygulandığında aşağıdaki süreç adımları ortaya çıkar (Tompkins ve White, 1984):

- Tesisin amacını tanımla (veya yeniden tanımla): Hem bir üretim tesisinde hem de bir hizmet işletmesinde, üretilecek ürünün ya da sunulacak hizmetin niceliksel olarak belirtilmesi gerekmektedir. Hacimler ya da etkinliklerin düzeyleri olanaklı ise tanımlanmalıdır.
- Amaç gerçekleştirilirken ana ve destek etkinlikleri tanımla: Yürütülen ana ve destekleyici etkinlikler ve karşılanması gereken gereksinimler; operasyonlar, donanım, personel ve malzeme akışları dikkate alınarak tanımlanmalıdır.
- Bütün etkinlikler arasındaki ilişkileri belirle: Tesis sınırları içerisinde etkinliklerin birbirlerini nasıl etkiledikleri ya da destekledikleri belirlenmelidir. Hem niceliksel hem de niteliksel ilişkiler tanımlanmalıdır.
- Bütün etkinlikler için alan gereksinimlerini belirle: Her bir etkinlik için alan gereksinimi belirlenirken, tüm malzemeler, donanım ve personel gereksinimi de gözönünde bulundurulmalıdır.
- Seçenek tesis planları oluştur: Seçenek tesis planları hem seçenek tesis kuruluş yerlerini hem de seçenek tesis tasarımlarını içermektedir. Seçenek tesis tasarımları, seçenek yerleşim yeri tasarımlarını, yapısal tasarımları ve malzeme taşıma sistemlerini içermektedir.
- Seçenek planları değerlendir: Her aday tesis planı için, içerdikleri öznel etmenleri belirle ve bu etmenlerin tesisi ve operasyonları etkileyip etkilemediğini ve ne kadar etkilediğini değerlendir.
- Bir tesis planı seç: Hangi aday plan, tesisin amaç ve hedeflerini en iyi ölçüde sağlıyorsa onu seç.

- Seçilen tesis planını uygula: Plan seçildikten sonra, bir sonraki adım onu gerçekleştirmektir. Bunun için analistleri sorumlu tutmak iyi bir düşünce olabilir. Bilindiği gibi, çalışanlar değişime genelde direnç gösterirler. Eğer yeni bir tesis düzeni oluşturulmuşsa bunun öncelikle çalışanlar tarafından onaylanması ve desteklenmesi gerekir (Heragu, 1997).
- Uygulanan tesis planını sürdür ve uyumlandır: Yeni gereksinimler tesis üzerinde yerleştirildikçe, tüm tesis planı da güncellenmelidir. Ürün tasarımındaki değişimler, taşıma ekipmanlarında veya akış modellerinde değişiklik yapılmasına dolayısıyla da tesis planında güncellemeye gidilmesine gereksinim doğurabilir.

2.2.1.2. Gezi Şeması

Makineler ve bölümler arasında malzemelerin, işgörenlerin veya araç-gereçlerin belli bir zaman dönemi içindeki hareketlerini sayısal olarak gösteren çizelgelere gezi şeması denir. Yerleşim düzenlenmesi tasarımında kullanılır (Erkut ve Baskak, 2003).

2.2.1.3. Atama yöntemi

Atama yöntemi, n tane bölümün m tane alana atanmasını minimum maliyetlerle gerçekleştirmeye çalışan bir yöntemdir. Atama yöntemi, tesis içinde makinelerin, makinelere işlerin ve işgörenlerin atanması için de kullanılabilen bir yöntemdir.

2.2.1.4. Wimmert Yöntemi

İlk sayısal teknikleri önerenlerden biri olan R.J.Wimmert ölçek olarak makine veya bölümler arasındaki uzaklık-hacim-malzeme taşımalarının en küçük olmasını kullanmaktadır. Üç temel varsayımı vardır (Erkut ve Baskak, 2003):

- Bölüm yerleri birbiriyle yer değiştirebilir,
- İki yer arasındaki uzaklık hareketin yönüne bağlı değildir,
- Maliyet ile eşdeğer uzaklık arasında doğru orantı vardır.

2.2.1.5. Spiral Analiz Yöntemi

Spiral analiz yönteminin amacı, imalat bölümlerinin sıralamasında birbirini izleyen işlemler arasında en dolaysız malzeme akışını sağlamaktır. İlk yapılan yerleştirmenin rasyonel olduğu varsayımı altında fabrika yerleştirilmesinde değişiklikler yapılması durumunda, bölümlerin gereksinmesi açısından çok küçük değişiklikler söz konusu olacaktır. Bundan başka bölümler veya atölyelerin kare veya dikdörtgen binalardan oluşması nedeniyle binaların çevresel biçiminde yapılacak değişiklikler sonucunda

imalat bölümlerinin kapladığı alanlar açısından ortaya çıkacak etkiler çok küçük olacaktır. Bu iki varsayım altında temel amaç, eldeki toplam alan içinde, var olan birim alanların konum ve biçim açısından ilişkilerini belirlemektir (Erkut ve Baskak, 2003).

2.2.1.6. *Hat Dengeleme*

Ürün oluşumu sırasında yapılması gereken işlerin, montaj istasyonlarına, kayıp süreleri en aza indirecek şekilde atanması işlemine hat dengeleme denir. Hat dengeleme kavramı yerleşimler arasında kıyaslama yapılmasını sağlayacak denge kaybı vb. sayısal sonuçlar üretebildiğinden yerleşim tasarımında kullanılacak yöntemlerden biri olmaktadır.

2.2.1.7. *Doğrusal Hat Yerleştirme Yöntemi*

Yöntem; makinelerin veya bölümlerin düz doğrular üzerinde yer aldığı ve malzeme akışlarının da bu doğrular üzerinde gerçekleştiği varsayımına dayanır. Eğer çaprazlama atlayışlar olabilecekse yöntem bu tip yerleştirme problemlerinde kullanılamaz (Erkut ve Baskak, 2003).

2.2.1.8. *Dal-Sınır Algoritması*

Dal-sınır algoritması, göz önünde bulundurulan optimizasyon probleminin tüm çözümlerinin örtülü sayımına dayanır. Arama alanı, problemin çözümünü ve onun bütün ilgili arama alanını gösteren kök düğümlere sahip dinamik bir ağaç yapı tarafından keşfedilir. Yaprak düğümler potansiyel çözümlerdir ve iç düğümler tüm çözüm uzayının alt çözümleridir. Arama ağacının budanması herhangi bir sonuç elde edemeyen alt ağaçların budanması işlemi olan sınırlama fonksiyonuna dayanır. Dal-Kesme yöntemi dal-sınır metodu ile çok yüzlü metodun birleşimiyle oluşmaktadır. Bu metotla büyük boyuttaki bazı zor problemler başarı ile çözülmüştür (Talbi, 2009).

Bazaraa makalesinde, tesis düzenlenme problemi için bir kuadratik küme örtülü formülasyonu önermiştir. Formülasyon hem tek katlı hem de çok katlı; düzgün veya düzgün olmayan şekilli bölümlere sahip yapılar için; bir yerleşimi en baştan tasarlamak veya var olan bir yerleşime yeni bir nesne eklemek için kullanılabilir. Nihâ yerleşim, bir dal-sınır algoritması yordamı ile elde edilmiştir.

2.2.1.9. *Montreuil's Karma-Tamsayılı Programlama Algoritması*

Montreuil, ayrık uzay yerine sürekli uzayda yerleşim düzenlenmesi optimizasyonu sunan modellerin kapsamlı bir serisini tanıtmıştır. Modelde her merkez için girdi-çıkı

istasyonlarını entegre etmeyi tanımlamıştır. Bu modeller karma tamsayılı doğrusal programlardır (Montreuil vd., 2004)

Tesis yerleşimi düzenlenmesi probleminde optimal çözüm elde etmek için “teoride” iki geleneksel yaklaşım daha geliştirilmiştir. “Teori” ifadesi kullanılmasının nedeni her iki yaklaşımın da genellikle optimallığı garanti edememesidir. Bunlardan ilki karesel atama problemi yaklaşımı, diğeri ise grafik teorisi yaklaşımıdır (Meller, 1996).

2.2.1.10. Grafik Teorisi Yaklaşımı

Grafik Teorisi Yaklaşımında, birbirine komşu tesis çiftlerinin yerleşim önemlerinin bilindiği varsayılır. Bölümlerin alanları ve şekilleri ilk başta yok sayılır ve her bölüm grafikte bir nokta olarak gösterilir. Bölümlerin uygun komşuluk ilişkileri grafikte bitişik bölümleri birleştiren yaylarla gösterilir. Amaç fonksiyonu, bölümler arasındaki komşuluğun oranının maksimum olduğu bir grafik düzenleyen maksimizasyon denklemidir (Meller, 1996).

Bu yaklaşımda, tesis çiftlerinde her birinin diğesine bitişik olmasının istendiği varsayılır. Bu modelde, i tesisini j tesisine bitişik yerleştirmenin isteğini belirten bir yakınlık derecesi kabul edilmiştir. Model, tesislerin yakınlık derecelerini maksimum kılmaya çalışır. Rosenblatt malzemelerin taşıma maliyetini minimum kılan ve yakınlık derecelerini maksimum kılan bir model geliştirmiştir. Bu modelde problemin çözümü için bir sezgisel algoritma geliştirmiştir (Shouman v.d., 2001).

2.2.1.11. Karesel Atama Problemi

Karesel Atama Problemi (KAP) Koopmans ve Berkman tarafından eşit alanlı tesislerin etkileşimini bulma problemi olarak tanıtılmıştır. KAP kentsel planlama, kontrol paneli yerleşimi ve kablolama tasarımını içeren çok geniş bir uygulama alanında kullanılır. KAP taşıma mesafesini minimum yapmaya çalışan tesis yerleşimi düzenlenmesi probleminin özel bir durumudur., çünkü tüm bölümlerin eşit alanlara sahip olduğu (veya bir yerden diğesine uzaklık önceden belirlenebilir ve bölümlerin atamalarını dikkate almadan aynı kalır) ve bütün yerlerin sabit olduğu ve yerleşimlerin önceliklerinin bilindiği varsayılır (Meller, 1996).

İlk olarak Koopmans ve Beckman yerleşim alanları arasındaki malzeme akışı ile alanların yerleşim problemini karesel atama problemi olarak modellemişlerdir. Bu ismi alma sebebi; amaç fonksiyonu değişkenleri ikinci derece olan bir fonksiyondur ve kısıtları değişkenlerin doğrusal fonksiyonlarıdır. Alanlar arası uzaklık ve akışla ilişkili maliyeti minimum yapma amacı ile bir grup tesisin bir grup yerleşim alana yerleştirilmesi problemi göz önüne alınır.

KAP, TYDP çözümünde sıklıkla kullanılan bir modeldir. Ancak bu her TYDP'nin KAP ile çözülebileceği anlamına gelmez (Shouman vd., 2001).

Tesis düzeni ve yerleştirme sorunlarının kimileri, var olan tesis düzeninde tek yeni bir alanın yerleştirilmesi ya da tek bir bölüm için tesis tasarımını içerir. Tesis düzeni ve yerleştirilmesi sorunlarının çözülmesinde nicel öğelerin yanı sıra nitel öğelerin de göz önüne alınması çok güç bir iştir. Bu nedenle nicel ve nitel öğelerin, seçenekli çözümlerin yerleri değerlendirilebilmesi için nicel biçimde birleştirilmesi zor olmaktadır. Genellikle, kimi tesis düzeni ve yerleştirme sorunlarının nicel amaç kullanılarak çözüldüğü, daha sonra çözümün nitel incelemelere dayanılarak değiştirildiği gözlemlenir. Normal koşullar altında bu süreç oldukça yeterlidir ve nitel incelemeleri göz önüne almayan bir yaklaşımdan daha üstündür (Erkut ve Baskak, 2003).

2.2.2. Sezgisel Algoritmalar

Bir çok uygulamada, bir problemin optimum olmasa da optimuma yakın makul bir çözümünü üreten, fakat bu çözümün optimumluğunu garanti etmeyen algoritmalar vardır. Çözüm belki optimumu elde edebilir fakat bunun matematiksel olarak ispatı yapılamaz. Bu nedenle, sezgisel algoritmalar optimuma yakın çözümler verecek şekilde tasarlanır. Sezgisel algoritmalar, problemin optimum çözümünün bilinen bir yöntemi olmadığı ya da bilinen bir yöntem olmasına rağmen çözüm zamanının ekonomik olmadığı ve yakın optimal çözümlerin tatmin edici olduğu durumlarda kullanılır. Sezgiseller, geniş boyutlu örneklerde iyi sonuçlar bulurlar. Problemlerin geniş bir aralığında, kabul edilebilir bir maliyette kabul edilebilir performans elde etmeye izin verirler. Özel sezgiseller ve meta sezgiseller şeklinde iki gruba sınıflandırılabilirler. Özel sezgiseller özel bir problemi ya da örneği çözmek için tasarlanırlar. Meta sezgiseller, hemen hemen

her optimizasyon problemine uygulanabilen genel amaçlı algoritmalarıdır. Bunlar, özel optimizasyon problemlerini çözmek için tasarlanan sezgisellerde yol gösterici strateji olarak kullanılabilen üst düzey genel yöntemler olarak görülebilir.

Kesin algoritmaların çalışması verimli olmayabilir. Bu durumda, ideal çözüme yakın bir çözümü, kesin algoritmaları kullanmadan da bulabiliriz. Pratikte sezgisel algoritmalar, kesin algoritmalara tercih edilmektedir (Aytekin ve Kalaycı, 2010).

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

Algoritmanın her bir iterasyonunda, uygun çözüm gibi bir permütasyon kullanan kurma yöntemi Gilmore tarafından 1962'de 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. Bununla birlikte, en iyiyi garantilemek çok daha fazla zaman alır.

İyileştirme yöntemleri, yerel arama algoritmaları ile benzerlik gösterirler. KAP 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 kalmayınca kadar devam eder. Bu yöntemin temel elementi komşuluk ve komşulukların hangi sırayla analiz edileceğini belirleyen seçim kriteridir.

Yiğit ve Türkbey(2003), sezgisel ilk çalışma olarak Kuehn ve Hamburger'in yapmış olduğu yerleşim problemlerinin büyük bir kısmını tasarlamaya yönelik sezgisel yaklaşımını belirtmişlerdir. Bu yaklaşım iki adımdan ibarettir. Birinci adımda, amaç fonksiyonunu artırarak en çoklayacak şekilde tesisleri sırasıyla açmakta ve amaç fonksiyonu değerini düşüren yeni bir tesis eklendiğinde durmaktadır. Kuehn ve Hamburger bu adımı "adım ekleme" olarak isimlendirmişlerdir. Literatürde ise, bu yöntem her bir adımında azami getiri hedeflediğinden dolayı "açgözlü sezgisel" olarak adlandırılmaktadır. İkinci adımında ise "yok et ve adım değiştir" kuralı bulunmaktadır. Bunun sebebi, ekonomik olmayan aday tesisi yokedip açgözlü sezgiselle altsırada seçilmiş olan tesislerden birini değiştirmektir. Sonra da bu olurlu çözümden başlayıp açık tesisle kapalı tesisi değiş-tokuş yaparak çözüme gitmektedir. Bu sezgisel algoritmada "içdeğişim sezgisel" olarak adlandırılmaktadır. Açgözlü ve içdeğişim sezgiselleri birçok yaklaşım algoritmasının temelidir. Her ikisi de gerekli başlangıç olurlu çözümü veya olurlu çözümleri kullanarak kesin algoritmalara yardımcı olmaktadır. Bununla birlikte kullanıcının sezgisel çözüm değerinin optimal çözüm değerinden uzaklaşmadığı noktada tatmin olması için, nihai çözümde sezgisellerin kullanımında üst sınır olmalıdır (Yiğit ve Türkbey, 2003).

2.2.3. Meta Sezgisel Yaklaşımlar

Meta sezgisel yöntemler, çözüm alanının en elverişli bölgelerinin derin aramasını gerçekleştirmeye dayalı yöntemlerdir. Meta sezgisel yöntemlerle üretilen çözümlerin kalitesi, klasik sezgisellerle elde edilenlere göre daha yüksektir. Meta sezgiselleri çözüm yaklaşımları açısından temel olarak üç ana grupta toplayabiliriz; yerel arama (örn; tavlama benzetim, deterministik tavlama, tabu arama), popülasyon arama (örn; uyarlamalı hafıza metotları, genetik arama), öğrenme mekanizmaları (yapay iletişim ağları, karınca kolonisi sistemleri). Laporte tarafından yapılan araştırmaya göre, meta sezgisel algoritma sınıfına giren çözüm yöntemleri ve literatür aşağıdaki gibi verilmiştir (Tezer, 2009):

- İlk tabu arama uygulaması (Willard, 1989)
- Taburoute" un ilk versiyonu (Gendreau, Hertz, Laporte, 1991)
- Tabu arama (Taillard, 1993)

- Tavlama benzetimi (Simulated Annealing) ve tabu arama (Osman, 1993)
- Taburoute (Gendreau, Hertz, Laporte, 1994)
- Uyarlamalı bellek (Adaptive memory) (Rochat, Taillard, 1995)
- Çıkarma zincirleri (Ejection chains) (Rego, Roucairolx, 1996)
- Birleştirilmiş tabu arama (Cordeau, Laporte, Mercier, 2001)
- Uyarlamalı bellek (Adaptive memory) (Tarantilis, Kiranoudis, 2002)
- Tanecikli (Granular) tabu arama (Toth, Vigo, 2003)
- Çok büyük komşuluk arama (Ergun, Orlin, Steele-Feldman, 2003)
- Belirleyici tavlama (Deterministic annealing) (Li, Golden, Wasil, 2004)
- Popülasyon arama (Prins; Mester and Bräysy, 2004)
- Karınca sistemi optimizasyonu (Reinmann, Doerner, Hartl, 2004) .

2.2.3.1. Değişken Komşuluk Arama

Değişken Komşuluk Arama (Variable Neighborhood Search – DKA), komşuluk yapılarının sistematik biçimde değiştirilmesi esasına dayanan bir metasezgiseldir. 1997 yılında Pierre Hansen ve Nenad Mladenovic tarafından geliştirilmiştir (Hansen ve Mladenovic, 1997).

Değişken komşuluk arama yaklaşımı aşağıdaki düşüncelere dayanmaktadır:

- Bir komşuluk yapısına göre lokal minimum olan, başka bir komşuluk yapısına göre de yerel minimum olmak zorunda değildir
- Bir global optimum tüm komşuluk yapılarına göre yerel optimumdur.

Bir çok problem için yerel minimum, bir veya birkaç komşuluğa göre birbirlerine yakındır (Hansen ve diğerleri, 2006).

2.2.3.2. Parçacık Sürü Optimizasyonu Algoritması

Parçacık Sürüsü Optimizasyonu (PSO) algoritması, ilk kez Kennedy ve Eberhart tarafından 1995 yılında ortaya atılmış, kuşların sürü hareketlerinden esinlenilmiş bir “metaheuristik” (Komşu çözümlerle haberleşerek iterasyonlarla en iyiye yakın sonucu veren) algoritmadır. Bu tip algoritmaların temsilcileri Karınca Kolonisi Optimizasyonu, Genetik Algoritmalar, Yapay Sinir Ağları, Evrimsel algoritmalar, Benzetimli Tavlama, Tabu Arama olarak sayılabilir. Temel prensibi, sürünün birer bireyi olan parçacıkların,

kendilerinin ve birbirlerinin deneyimlerinden yararlanarak bilinen en iyi sonuca doğru yönelmeleridir (Önüt vd., 2007).

Koloni veya sürü halinde yaşayan sosyal böcekler ve diğer canlılar herhangi bir otorite olmadan çalışırlar. Bu tip yapılarda takım çalışması bireyler arasındaki çeşitli etkileşimler sonucu kendi kendine ortaya çıkan bir unsurdur. Bu etkileşimler ne kadar basit olursa olsun bir araya geldiklerinde büyük problemler için etkin çözümler sunabilirler. Sürü Zekası algoritmaları balık, kuş, arı gibi sürü halinde hareket eden canlıların sosyal düzenlerini modelleyen yaklaşımlardan esinlenilerek geliştirilmiş algoritmalarıdır. Bu algoritmaların iki temel özelliği herhangi bir liderin olmaması ve sürü içindeki etkileşimlerin yerel olmasıdır. Bu iki özellik nedeniyle, geliştirilen modellerde komşuların hızına uyma, komşularla çarpışmaktan kaçınma ve komşulardan çok uzaklaşmaktan kaçınma gibi davranışlar ortaya çıkmaktadır (Küçükdeniz, 2009).

Bu tekniğe kaynak oluşturan sosyal benzetim şu şekilde özetlenebilir: Bir toplumun bireyleri, bir inanç uzayının (çözüm uzayı) parçası olan ve her mümkün birey tarafından paylaşılan fikirlere sahiptir. Bireyler bu fikir kavramını faktöre bağlı olarak değiştirebilirler (Cura, 2008):

- Çevre bilgisi (uygunluk değeri)
- Bireylerin geçmişe yönelik deneyimleri (belleği)
- Komşuların geçmişe yönelik deneyimleri

Literatürde PSO kullanılarak çalışılmış çok fazla tesis yerleşimi ve KAP uygulamaları olmasa da Liu ve Abraham (2007) tarafından yapılmış melez-bulanık komşuluk yapısını kullanan PSO algoritması ile KAP problemlerinin çözümü üzerine bir çalışmayı; Rezazadeh vd. (2009) tarafından çalışılmış PSO ile dinamik tesis yerleşimi düzenlenmesini; Önüt vd. (2007)'nin çalışması olan PSO ile çok katlı tesis yerleşim düzenlenmesini ve Zhang ile Wang (2007)'a ait olan SO ile eşit olmayan alanlara sahip tesis yerleşimi düzenlenmesi çalışmasını örnek verebiliriz.

2.2.3.3. Karınca Kolonisi Optimizasyonu Algoritması

Karıncalar, kolonilerinin menfaatleri için beraber çalışan sosyal böceklerin en iyi örneklerindedir. Koloni halinde yaşayan karıncalar yiyecek bulmak için ilk olarak öncü karıncaları tek başına gönderirler. Bu öncüler çevreyi araştırarak uygun yiyecek kaynağını bulmaya çalışır. Öncüler yiyecek bulursa, koloninin olduğu yere geri dönerken arkalarında feromon bırakarak ilerler. Bu feromon sayesinde diğer karıncalar da bu yiyecek kaynağını bulabilirler.

Aslında karıncalar yukarıda bahsedilenden çok daha karmaşık bir yöntemi uygularlar. Yiyecek kaynağını başarıyla bulan öncü karınca geri dönerken en kısa yoldan dönmüş olmayabilir, tabii bu da kolonidekilerin karmaşık yollarla kaynağa gitmesine neden olacaktır. Aynı yiyecek kaynağını keşfeden başka bir öncü karınca belki buraya daha kestirme bir yol bulmuş olabilir. Kolonidekiler hangi öncünün en kestirme yolu bulunduğunu nasıl bilecektir? Bu nedenle bazen koloniler gereksiz derecede uzun bir yoldan gitmek zorunda kalabilirler. Ama kestirme yollardaki feromonlar daha düzenli olarak yenilenir ve bu sayede de karıncalar daha belirgin feromon izi olan yani daha kısa yolu tercih ederek gereksiz derecede uzun yollardan ilerlemek zorunda kalmazlar (Alba vd., 2004).

Yapay karıncaların gerçek karıncalardan farklı olarak birkaç ekstra özelliği daha vardır. Bunların başında; geçmiş davranışlarını hafızada tutan veri yapılarına sahip olmaları gelir. Gerekliğinde “Yerel arama” gibi bir prosedür uygulanarak hesaplanmış yolların kalitesinin arttırılması üzerine gidilebilir. Çoğu durumda feromon güncelleştirilmesi yol üzerinde ilerlemeye devam edilirken değil ancak tüm yol yapılandırıldıktan sonra yapılır ve yol üzerinde biriken feromon genellikle tüm yolun kalitesinin bir fonksiyonudur. Sonuç olarak bir yapay karıncanın bir yolu seçmesinin olasılığı yalnız feromon miktarına değil aynı zamanda bazı probleme özgü yerel sezgilere de bağlıdır (Solnon, 2008).

Tüm karıncalar turlarını tamamladıktan sonra feromon miktarları güncellenir. İlk olarak tüm yollardaki feromonlar buharlaşma oranına göre buharlaştırılmaktadır. Daha sonra karıncaların geçtikleri yollardaki feromon miktarı yolun uzunluğuna göre

arttırılmaktadır. Yani o yolu kullanan karıncanın toplam yol uzunluğu fazlaysa feromon miktarı az arttırılacaktır. (Söyler, Keskinürk, 2007).

Literatürde KKO algoritması kullanılarak yapılmış bir çok tesis yerleşim çalışması ile karşılaşabiliriz. Merz ve Freisleben (1999)'a ait sezgisel yöntemlerin karşılaştırıldığı bir KAP çalışması; Gambardella vd. (1999)'nin çalıştığı KAP için bir KKO algoritması; Baykasoğlu vd.((2004)'ne ait dinamik tesis yerleşimi düzenlenmesi için çalışılmış bir KKO algoritması; Toksarı (2004) tarafından hazırlanmış bir KKO ile tesis yerleşimi çalışması ile Kendall ve Shang (2004) tarafından yapılan melez bir dinamik tesis yerleşimi çalışması örnek verilebilir.

2.2.3.4. Genetik Algoritma

Genetik Algoritmalar (GA), arama ve eniyileme problemlerini çözmekte kullanılan uyarlanabilir bir yöntemdir. Doğada bireyler yiyecek, su ve barınak gibi kaynaklar için yarış halindedirler. Aynı zamanda her birey soyunu devam ettirmek ister. İşte bu şartlarda, yarışı kazanan (güçlü ve çevreye en iyi uyumu gösteren bireyler) hem kaynaklara sahip olur hem de soyunu devam ettirme şansını elde eder. Yarışı kazanan bireylerin ürettiği yeni bireyler de, atalarından gelen özellikleri alırlar.

GA, yukarıda bahsedilen “doğal seçim ve uyarlanım” prensibinden esinlenilerek ortaya atılmış bir yaklaşımdır. Bireyler, GA’da ilgili problemin çözümlerini temsil etmektedir. Bireylerin (yani kromozomların) ortama uyum sağlama ve hayatta kalma durumu (yani bireyin “uygunluk değeri”) ise, GA’da, ilgili çözümün problemi çözebilme yeteneğini temsil etmektedir. GA mevcut birçok çözümden en uygunlarını seçerek, bu çözümlerden yeni çözümler elde etmeyi amaçlar (Gülsün vd., 2009).

GA, popülasyon temelli sezgisel bir optimizasyon tekniğidir. Özellikle doğrusal olmayan çok değişkenli problemlerin çözümünde etkin olarak kullanılmaktadır. Çözüm alternatifleri kromozom denem yapılarına kodlanmaktadır. Her bir kromozom bir çözüm alternatifini temsil etmektedir. Seçimle daha iyi çözümlerin elde tutulması sağlanırken, çaprazlama ve mutasyon denem operatörlerle kromozomlarda değişiklikler yapılmakta ve alternatif çözümler aranmaktadır. Amaç iterasyonlar boyunca yapılan değişikliklerle en uygun sonuca yaklaşımdır (Er, Keskinürk, 2007).

GA'nın en önemli avantajı paralel çalışmasıdır. Büyük problemler için bu önemli bir kazançtır. Çok geniş bir çözüm uzayını hızlı bir şekilde arayabilir. Geniş bir problem aralığında kullanılabilirler. Başlangıçta problem hakkında bilgi olmadan da problemleri çözebilir. GA'nın gerçekten iyi çalışması için olası çözümlerin iyi temsil edilmesi, uygunluk fonksiyonunun, işleçlerin ve parametrelerin iyi belirlenmesi başarı için şarttır. GA'da yerel en iyi değer yakınsamanın bile garantisi yoktur. GA'da olduğu bilinen en önemli problem erken yakınsamadır. Bunun anlamı diğer bireylerden çok daha iyi olan bir birey tüm çözüm arama sürecini etkisi altına alarak, gerçek çözüme (global optimum) yaklaşılmasını engelleyebilir, yerel en iyi değerde (local optimum) takılıp kalınmasına neden olur. GA'nın dezavantajlarını aşmak için yerel arama yöntemleri ve diğer eniyileme yöntemleriyle melez yöntemler geliştirilmektedir (Gülsün vd., 2009).

GA da tüm diğer sezgisel yaklaşımlar gibi rastlantı aramaya dayanmaktadır. GA'nın öncülüğünü 1960'larda John Holland yapmıştır. O günlerden bu yana birçok araştırmacı tarafından çalışılmış, geliştirilmiş ve mühendislik başı çekmek üzere pek çok uygulama alanı bulmuştur. Yalnızca alternatif bir yöntem olmakla kalmamış, literatürdeki pek çok çalışmada diğer sezgisel tekniklere üstün geldiği de gözlenmiştir (Cura, 2008).

Bu çalışmalara örnek olarak, Singh ve Sharma (2005) tarafından tesis yerleşimi problemine farklı yaklaşımların incelenmesi; Balakrishnan vd. (2002)'ne ait melez bir GA ile dinamik tesis yerleşimi düzenlenmesi çalışması; Mak vd. (1998)'nin GA ile tesis yerleşimi çalışmaları; Lee vd. (2003)'ne ait gelişmiş bir GA algoritması kullanılarak yapılmış bir tesis yerleşimi çalışması; Lee vd. (2005) çalışmalarını geliştirip çok katlı tesis yerleşimi problemlerine GA uygulamaları ve yine tesis yerleşimi düzenlenmesine uygulanmış GA çalışmalarına sahip Azadivar ve Wang ile Li ve Love (1998) örnek verilebilir.

2.2.3.5. Yapay Sinir Ağları

Yapay Sinir Ağı, insan beyni gibi biyolojik sinir sisteminden ilham edilen bir bilgi işlem paradigmasıdır. Bu paradigmanın anahtar elemanı, bilgi işlem sisteminin yapısıdır. Belirli problemleri uyum içinde çözmek için işlem elemanları arasında yüksek sayıda bağlantı oluşturulur. Yapay sinir ağları insanlar gibi örneklerle öğrenir. Bir yapay

sinir ağı, öğrenme işleme içerisinde veri sınıflandırma, veya numune tanıma gibi kesin (belirli) uygulama için şekillendirilir. Öğrenme, sınırlar arasında var olan bağlantılarda biyolojik sınırlardaki kuralları gerektirir (Fırat ve Güngör, 2004)

İnsan beyin fonksiyonundan esinlenen yapay sinir ağları (YSA), deneme yolu ile öğrenme ve genelleştirme yapabilmektedir. YSA'nın kullanıldığı önemli alanlardan biri de geleceği tahmindir. YSA, veriler arasındaki bilinmeyen ve fark edilmesi güç ilişkileri ortaya çıkartabilir. YSA doğrusal değildir. Doğrusal modeller, önemli detayları anlayabildikleri ve açıklayabildikleri takdirde avantajlı olabilirler. Ancak ilgilenilen problemin temelindeki ilişki doğrusal olmadığı durumlarda doğrusal modeller uygun değildir. YSA, girdi ve çıktı değişkenleri arasındaki herhangi bir ön bilgiye ihtiyaç duymadan, herhangi bir varsayımda bulunmadan, doğrusal olmayan modellemeyi sağlayabilmektedir. Ağa, girdi bilgileri ve bu girdilere karşılık gelen çıktı bilgileri verilmekte ve ağın girdi-çıktı arasındaki ilişkiyi öğrenmesi sağlanmakta, böylece ağın eğitimi gerçekleştirilmektedir. Öğreticili öğrenme olarak adlandırılan bu yöntem genelde tercih edilen bir yöntemdir (Hamzaçebi ve Kutay, 2004).

Literatürde tesis yerleşimi düzenlenmesinde yapay sinir ağlarının kullanımı yaygın olmamasına rağmen bu konuda çalışmalar da yapılmıştır. Tsuchiya vd. (1996) bir karesel atama probleminin çözümü için yapay sinir ağları yaklaşımını kullanmış, Benjamin vd. (1995) ise yapay sinir ağlarının tesis yerleşimi problemindeki performansını göstermek için BP ve ART II paradigmasını karşılaştırarak bir çalışma yapmışlardır.

2.2.3.6. Tavlama Benzetimi Algoritması

Adından da anlaşılacağı üzere Tavlama Benzetimi (TB) bir metalin soğuyarak ve donarak minimum enerjili kristal yapısına dönüşmesi (tavlama süreci) ile daha genel bir sistemde minimumun araştırılması arasındaki benzerlikten yararlanır. TB yaklaşımını daha iyi anlayabilmek için tavlama hakkında kısa bir bilgi sahibi olunması gerekmektedir. Herhangi bir katı madde erime noktasını aşınca kadar ısıtılır ve ardından katılaşınca kadar soğutulursa, bu katı maddenin yapısal özellikleri soğuma hızına bağlı olarak değişir. Örneğin büyük kristaller çok yavaş soğutulacak olursa gelişmeler gözlenebilirken; hızlı soğuma neticesinde yapılarında birçok bozulmalar

barındırır. Görüldüğü gibi ısıtılan ve ardından belli bir hızla soğutularak en iyi biçime ulaştırılmaya çalışılan madde, bir sistemdeki parçacık gibi algılanırsa bu tavlama sürecinden TB elde edilmiş olur (Cura, 2008).

TB, ilk olarak Kirkpatrick tarafından optimizasyon problemleri için kullanılmaya başlanmıştır. TB yerel optimumdan kurtulabilme özelliğine sahip olan olasılıklı bir arama algoritmasıdır ve pek çok farklı alana yaygın olarak uygulanmaktadır. Kolay uygulanabilirliği, yakınsama özelliği ve yerel minimumlardan kurtulabilme özelliği nedeniyle son yıllarda yaygın bir yaklaşım olmuştur. TB algoritması adım adım iyileştirme yapan bir algoritmadır. Bu iyileştirmeler esnasında, sadece iyi çözümler değil aynı zamanda belli bir olasılık ile kötü çözümleri de kabul eder. Bu olasılığa kabul olasılığı denir. TB diğer sezgisel yöntemlerle (bunlar genetik algoritma, tabu arama ve yapay sinir ağları vb.) karşılaştırıldığında bazı güçlü yanlara ve zayıf yanlara sahiptir. TB'nin güçlü yanları olarak diğer yöntemlere göre problemlere daha kolay uygulanabilir olması ve pek çok kombinatoriyal optimizasyon problemleri için iyi sonuçlar sağlama yeteneğidir. TB'nin zayıf tarafları ise, problemlerin çözümü için yüksek bilgisayar zamanına ihtiyaç duyması ve parametre seçiminin dikkatli bir şekilde yapılması ihtiyacıdır. TB'nin performansı büyük oranda parametrelerine bağlıdır. Bu nedenle TB için uygun parametrelerin belirlenmesi önemli bir konudur (Şahin ve Türkbey, 2010).

Uygulama kolaylığı, yakınsama özelliği ve yerel en iyilerden kaçınmak için kullandığı strateji nedeniyle TB, son yıllarda optimizasyon problemlerinin çözümünde kullanılan yaygın bir yöntem olmuştur. TB adım adım iyileştirme yapan bir yöntemdir (Şahin, 2008).

Literatürde Tavlama Benzetimi algoritmasının tesis yerleşimi düzenlenmesi için uygulandığı çalışmalar görebiliriz. Örneğin, Singh ve Sharma (2005) ile Yiğit ve Türkbey (2003) farklı yaklaşımlarla tesis yerleşim problemi üzerinde çalışmışlardır.

2.2.3.7. Tepe Tırmanma Algoritması

TT metodu bir iteratif iyileştirme (yerel arama) yöntemidir. TT, iniş ya da adım adım iniş strateji olarak da adlandırılmaktadır. Bu metodun temelinde, tanımlanan bazı

kurallara göre bir çözümden bir diğer komşu çözüme ulaşma vardır. TT metodunun etkinliğinde iyi bir komşuluk yapısı seçiminin önemi büyüktür. İyi mutlak açıdan en iyi olmak zorunda değildir. TT'nin zayıf yanı yerel ve genel en iyi arasındaki ayrımı yapamaması sonucu yerel optimumdan kaçamamasıdır. Yerel optimumdan genel optimuma geçebilmek için modern sezgiseller geliştirilmiştir. Yiğit ve Türkbey (2003) tarafından yapılan sabit kapasiteli tesis yerleşim probleminin çözümü için tepe tırmanma algoritması ve tavlama benzetimi algoritmasının karşılaştırıldığı çalışma örnek verilebilir.

2.2.3.8. Tabu Arama Algoritması

TA'da temel yaklaşım, son çözüme götüren adımın, dairesel hareketler yaratmasını engellemek için bir sonraki döngüde tekrarın yasaklanması veya cezalandırılmasıdır. Tabu yöntemi kısmen, benzer olaylarda raslantılı işlemler gerçekleştirirken insan davranışının tutarsızlığa eğilimli olmasından etkilenmiştir. Söz konusu işlem yeni rotaların rastlantılı olarak seçilmemesi dışında bu şekilde işler. Başka bir ifadeyle tabu arama zaten incelenmiş bir yol olmadığı sürece her çözümü araştırabilen bir süreçtir. Böylece yeni çözüm uzayının incelenmesi suretiyle yerel minimumdan kaçınılarak istenilen çözüme ulaşılabilir. Algoritma yerel minimuma doğru hareket ederek başlar. Daha önce yapılmış hareketlere tekrar dönüş yapmayı engellemek için yöntem bir veya daha fazla tabu listesi tutar. Listenin orijinal amacı, önceden yapılmış bir hareketin tekrarından çok tersine dönmesini önlemektir. Tabu listesi kronolojik bir yapıya sahiptir ve tabu arama belleğini biçimlendirir. Belleğin rolü algoritma ilerledikçe değişebilir. Başlangıçta hedef, çözüm uzayında kaba araştırma yapmaktır. Aday konumlar belirlendikçe arama yerel optimum çözümü üretmeye daha fazla odaklanır (Cura, 2008).

Tabu aramanın temel iki unsuru, daha önce denenmiş çözümleri belirli bir süre yeniden işleme almaması ve bu sayede yerel optimum noktadan uzaklaşıp küresel optimum çözüme ulaşabilmesidir.

Tabu Arama Yöntemi, üçüncü bölümde daha ayrıntılı bir şekilde incelenecektir.

2.3. BİLGİSAYAR DESTEKLİ TESİS PLANLAMASI

Tesis yerleşimi çalışmasında bölüm veya iş merkezi sayısının artması ile mümkün seçeneklerin sayısı çok daha fazla bir oranda artmaktadır. Örneğin hiçbir kısıt konulmadığı takdirde 10 iş merkezinin yerleştirilmesi için $10! = 3.628.800$ adet seçenek söz konusudur. Çok iş merkezli problemlerin grafik yoldan çözümü oldukça zor olduğundan, bilgisayar destekli tesis yeri düzenlenme yöntemleri geliştirilmiştir. Söz konusu yöntemler; yerleştirdiği bölüm veya iş merkezi sayısı, yerleştirdiği bina katı sayısı, bilgisayar işlem süresi ve sonuç kalitesi açısından farklı özelliklere sahiptir (Tanyaş, 2000).

Tanyaş (2000) bir tesisin düzenlenmesi için gerekli temel girdileri şöyle sıralamıştır:

- Bölümler veya işlemler kümesi,
- Bölümlerin alanları,
- Bölümler arası ilişkiler.

2.3.1. Tesis Yerleşimi Düzenlenmesi Problemi İçin Geliştirilen Temel Algoritmalar

Koopmans ve Beckmans'ın tesis düzenlenme ve yerleşim problemini 1957 yılında kuadratik atama problemi olarak modellemelerinden sonra, birçok algoritma önerilmiştir. Bunlar aşağıdaki gibi sınıflandırılabilir (Heragu, 1997):

- Eniyileme (optimal) algoritmaları
- Sezgisel algoritmalar
- Meta Sezgisel algoritmalar

Sezgisel algoritmalar da kendi içinde aşağıdaki gibi ayrılabilir (Heragu, 1997):

- Kuruluş algoritmaları
- Geliştirme algoritmaları
- Melez algoritmalar

2.3.1.1. Kuruluş algoritmaları

Bu algoritmalar, tesis düzenini taslak bir çözüm bularak kurarlar. Bu; tesis düzeni tasarımına ulaşıncaya değin bölümlerin birbirini izleyen seçimini ve yerleştirilmesini içerir. Kuruluş algoritmaları, belirli bir amaç ölçütüne göre, sıfır çözümden başlayarak

bölmeleri tek tek ele alıp yerleştirirler ve sonuçta, amacı gerçekleyen biçimde yerleşim düzeni planı (kalıp plan) oluştururlar.

Kuruluş algoritmalarına aşağıdaki örnekleri verebiliriz (Tanyaş, 2000):

- ALDEP (Automated Layout Design Programme - Makinalaştırılmış tesis düzeni tasarımı problemi)
- CORELAP (Computerized Relationship Layout Planning - Programlanmış tesis düzeni tasarımı programı)
- PLANET (Computerized Plant Layout Analysis and Evaluation Technique - Programlanmış tesis düzeni çözümlemesi ve değerlendirme tekniği)
- LAYOPT (Layout Optimizing Programme - Tesis düzenini eniyileştirme programı)
- CASS
- COLO2
- COMP1
- COMP2
- COMSBUL
- DOMINO
- GENOPT
- IMAGE
- KONUVER
- LAYADAPT
- LSP
- MST (Modified Spanning Tree Algorithm)
- MUSTLAP2
- PLAN
- RMA
- SISTLAP
- SUMI

Kurucu yöntemlerin en çok tanınanı olan CORELAP ilk kez 1967 yılında Robert C. Lee ve James M. Moore tarafından önerilmiş ve daha sonra R. Sepponen tarafından geliştirilmiştir. CORELAP yönteminde başlangıçta aşağıdaki veriler bilgisayara okutulmaktadır (Tanyaş, 2000):

- Bölümler (veya iş merkezleri) arasındaki ilişkileri gösteren ilişki matrisi,
- Bölüm sayısı,
- Bölümlerin sahip oldukları yerleşim alanları,
- İlişki matrisi girişleri için ağırlıklar

Tanyaş (2000) bu temel bilgilerin yanı sıra zorunlu olmayan aşağıdaki bilgilerin de bilgisayara okutulabileceğini belirtmiştir:

- Sonuç tesis yeri düzeninin ölçeği,
- Binanın veya arsanın uzunluk ve genişliği,
- Bazı bölümler için ön atamalar.

CORELAP yöntemi öncelikle her bölüm için toplam yakınlık derecesini hesaplar. Bölümler toplam yakınlık derecelerine göre büyükten küçüğe doğru sıralanır. En yüksek toplam yakınlık derecesine sahip bölüm yerleştirme yapılacak alanın merkezi noktasına yerleştirilir. Yöntem bundan sonra iki sorunun cevabını arar:

Bundan sonra hangi bölüm yerleştirilecektir? Bu bölüm “Başaran (Victor)” olarak adlandırılmaktadır.

Başaran bölüm yerleştirme yapılacak alana nasıl yerleştirilecektir? Yerleştirilen bölüm ise “Kazanan (Winner)” olarak adlandırılmaktadır.

Yöntem bir kazanan bölüm elde ettikten sonra bu bölüm ile A ilişkisi içinde olduğu Aday (Candidate) bölümler aramaktadır. Aday bölümler arasında en yüksek toplam yakınlık derecesine sahip bölüm Başaran bölüm olmakta ve düzene yerleştirme hakkı elde etmektedir. İlk kazanan ile A yakınlık derecesine sahip diğer aday bölümler de yerleştirildikten sonra yöntem ilk Başaran bölümü (ikinci olarak düzene yerleştirilen bölüm) Kazanan bölüm olarak nitelendirmekte bu bölüm ile A ilişkisi içinde olan aday bölümler toplam yakınlık derecesi büyüklüğüne göre yerleştirilmektedir. Daha sonra sırasıyla ikinci, üçüncü ve diğer başaran bölümler kazanan bölüm olarak nitelendirilmekte yine A ilişkisi aranmaktadır. A yakınlık derecesi kalmadıktan sonra yöntem başa dönerek sırasıyla B, C, D, E ve F ilişkileri için aynı işlemleri tekrarlamaktadır. Sonuçta tek bir tesis yerleşimi elde edilmektedir. Seçenek tesis

yerleşimi elde etmek için başlangıç verilerinin değiştirilmesi gerekmektedir. Kuruluş algoritmaları, yeni bir tesisin yerleştirilmesi sorununa çözüm getirebilirler (Tanyaş, 2000).

Bir diğer program, PLANET programıdır (Tompkins vd., 1996). 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 yöntemdir.

2.3.1.2. Geliştirme algoritmaları

Bu tür algoritmalarda öncelikle, tesisin var olan düzeni tümüyle incelenir ve bölümlerin yerleşimleri, tasarımı geliştirecek biçimde birbirleriyle değiştirilir. Daha doğrusu geliştirme algoritmaları; var olan yerleşim düzeninin planını girdi olarak ele alır ve amaç ölçütünün gerçekleştirilmesi doğrultusunda, dış sınırları aynı kalmak koşuluyla, plan üzerinde bölümlerin birbirleriyle yerlerini değiştirerek iyileştirilmiş bir yerleştirme düzeni planı elde ederler.

Geliştirme algoritmalarına örnekler şunlardır (Tanyaş, 2000):

- CRAFT (Computerized Relative Allocation of Facilities Technique -Tesislerin programlanmış görelî yerleştirilmesi tekniği)
- COFAD (Computerized Facility Design - Programlanmış tesis tasarımı)
- COSFAD
- GRASP
- OFFICE
- OPT Algorithm
- PREP
- COL (Computerized Model for Official Layout)

CRAFT yönteminde başlangıçta aşağıdaki veriler bilgisayara girilir (Tanyaş, 2000):

- Geliş-gidiş şeması verileri,
- Malzeme hareketi maliyet verileri (Birim yük ve uzaklık taşıma maliyetleri),
- Başlangıç tesis yeri düzeni.
- Yerleri değişmeyecek bölümler

CRAFT yönteminde bölümler arası toplam malzeme taşıma maliyetini en aza indirecek şekilde ikili bölüm yer değiştirmeleri gerçekleştirilir. Önce başlangıç tesis yeri düzeninin toplam malzeme maliyeti yük, uzaklık ve taşıma maliyeti verileri dikkate alınarak belirlenir. Daha sonra olası tüm bölüm yer değiştirmeleri saptanarak, en fazla malzeme taşıma maliyeti indirimi sağlayan değiştirme gerçekleştirilir. Bu şekilde hiçbir malzeme taşıma maliyeti indirimi sağlanamama haline kadar yöntem işlemeye devam eder. Her değişiklikte bölümlerin merkez noktası uzaklıkları yeni düzene göre yeniden belirlenir, seçilen malzeme taşıma teçhizatı özelliğine göre (söz konusu teçhizat farklı bölümler arasında farklı özelliklerde olabilir) malzeme taşıma maliyetleri hesaplanır ve yeni bir değişiklik ile toplam malzeme taşıma maliyetinin azalıp azalmadığına bakılır. En düşük toplam malzeme taşıma maliyeti değerine sahip tesis yeri düzeni yöntemin sonucudur. Bölümler arası yer değiştirmeler ikili veya üçlü bir şekilde yapılabilmektedir.

Ayrıca COFAD, ALDEP, CORELA yaklaşımları da yerleşim problemlerini bilgisayar ortamında çözebilmek için geliştirilmiş ilk programlardandır. CRAFT ve COFAD toplam malzeme taşıma maliyetini minimum kılmayı amaçlarken ALDEP ve CORELAP bölümlerin yakınlık oranlarını maksimum kılmayı amaçlar.

2.3.1.3. Melez algoritmalar

Birden fazla çözüm tekniğini kullanan algoritmalar, melez ya da karma algoritmalar olarak tanımlanırlar. Bu bölüm altında, melez algoritmalara BLOCPLAN örnek verilebilir. Blocplan, Donaghey ve Pire tarafından geliştirilmiştir. Yeni yerleşimin yapılandırılma ve var olan 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 hesaplar; uzaklığı esas alan amaç fonksiyonu ya da komşuluğu esas alan amaç fonksiyonu kullanabilir. Bölümleri bantlar şeklinde düzenler. Bantların genişliğini planlamacı değiştirebilir. Var olan yerleşim düzenleri üzerinde değişiklik yapabilmesi için bütün bölümlerin dikdörtgensele şekilde olması gerekir (Utkan, 2006).

3. MALZEME VE YÖNTEM

3.1. TABU ARAMA ALGORİTMASININ ORTAYA ÇIKIŞI

Glover ve Laguna (1997)'nin da değindiği gibi; telekomünikasyon, lojistik, finansal planlama, ulaşım gibi pratik düzenlenmelerde karşılaşılan zor optimizasyon problemlerinin çokluğu ve üretimi, güçlü optimizasyon tekniklerinin gelişimine sebep olmaktadır. Bu teknikler genellikle çeşitli araştırma alanlarından fikirlerin düzenlenmesinin sonucudur. Umulan, etkili ve bugünün optimizasyon problemlerinin karmaşıklığıyla başa çıkabilecek işlemlerin gelişmesidir. Her metodolojinin kısmi yapısını destekleyen fikirler bazen beklenmeyen bağlantılardan oluşur.

Şebeke akış problemi, elektrik ve suyun kullanım yöntemini paylaşır. Benzer şekilde benzetimli tavlama metalurjide fiziksel bir işleme dayanır. Genetik algoritma biyolojik fenomenlere dayanırken, karınca kolonisi sistemi ise ortak bir sorunun çözümünü canlandırır. Tabu Arama (TA)'nın felsefesi ise akıllı problem çözümü ilkelerinin bir derlemesinden oluşur. Bu anlamda, tabu aramanın yapay zeka ve optimizasyon alanlarının birleştirildiği seçilmiş görüşlere dayandığı söylenebilir.

TA'nın temel şekli Fred Glover tarafından tasarlanan fikirlere dayanır. Yöntem, genellikle engel olarak görülen yerel optimumluğun ya da esnekliğin sınırlarını geçmek için tasarlanmış işlemlere dayanır. Bu işlemin ilk örnekleri sistemli olarak ortaya konmuş ve aksi takdirde yasaklı olan bölgelerin araştırılmasına izin vermek için kısıtlamaları kaldırılmıştır.

Tabu arama; yerel optimumdan öte çözüm uzayını araştıran, yerel bir sezgisel arama işlemini yönlendiren bir meta-sezgiseldir. Yerel işlem, verilen çözümün komşuluğunu tanımlamak için "hareket" adı verilen işlemi kullanan bir aramadır. Tabu aramanın temel bileşenlerinden birisi daha esnek bir arama davranışı oluşturan uyumsal hafızayı

kullanabilmesidir. Hafıza tabanlı stratejiler bu yüzden tabu arama yaklaşımlarının ayırt edici özelliğidir.

3.2. TABU ARAMA ALGORİTMASINA GENEL BAKIŞ

Yöneylem araştırmasında “sezgisel” terimi, tam çözümü bulunması zor hatta imkansız olan problemler için makul çözümler bulmada kullanılan yöntemler için kullanılır. Özel olarak optimizasyonda bir sezgisel yöntem, optimale yakın ya da yaklaşık olarak optimal bir çözüme götüren (fakat bunu garanti etmeyen) stratejilere dayanan, hızlı ve pratik bir yöntem anlamına gelir. Buna göre, bu sezgisel yöntemlerden söz edildiğinde, “çözmek” fiili, “optimale, tatmin edici bir yakınlık bulmak” yan anlamına sahiptir. İyi sezgisel yöntemler, prensipte, izin verilen zamanda elde edilebilecek en iyi çözümü belirleyebilmelerine rağmen optimal bir çözüm bulmayı garanti etmezler. Bazı problemlerinse, optimal çözümleri için etkili nitelendirmeler sağlayan optimalite koşulları bilinmektedir ve bu nedenle bu problemler iyi problemler olarak da adlandırılmaktadırlar. İyi problemlerin çözümleri için geliştirilen tam algoritmalar da bu bilinen optimalite koşullarına dayanmaktadır. 1980’li yıllar ve sonrasında, karmaşık tam algoritmaların kullanımı, zamanın bilgisayar sistemleri sayesinde yaygınlaşmıştır.

İyi problemleri çözmek için kullanılan tam algoritmaların gelişimi, önemli bir araştırma başarısıydı. Buna karşın, yapılan bu çalışmalar sıfır-bir ve tam sayılı programlama problemleri ve kombinatoriyal optimizasyon problemleri gibi karmaşık problemler için makul zamanda optimal çözümü bulan herhangi bir tam algoritma üretememiştir. Birerleme adı verilen ve tüm olası çözümleri birer birer inceleyen yaklaşım kullanılarak, bilgisayar desteğiyle karmaşık problemlerin orta büyüklükte olanlarının çözülebilmesine karşın, genelde birerleme yaklaşımının zaman gereksinimi, çözülen problemin büyüklüğüyle üstel olarak büyüdüğünden, gerçek hayatta karşılaşılan problemlerin çözümünde etkisiz kalmaktadır. Bu nedenle büyük boyutlu, karmaşık problemlerin çözümünde, araştırmacılar sezgisel algoritmaların kullanımına yönelmek zorunda kalmıştır (Murty, 1995).

Tabu, “Bir şeyi veya bir kimseyi kutsal sayan, ona dokunulmasını, onun eleştirilmesini, onun kullanılmasını yasaklayan anlayış” olarak tanımlanmıştır. Tabu Arama algoritması

da bu tanımdan yola çıkılarak oluşturulmuştur. Optimizasyon problemlerinde ise, optimum çözümü elde etmek için kullanılan yasaklar ve kısıtlamaları ifade etmektedir (Değertekin vd., 2006)

Yöneylem araştırması alanında, tabu aramanın doğaüstü güçler ya da ahlaki konularla hiç ilgisi yoktur. Bunun yerine tabu arama, çözüm uzayında bazı bakımlardan güç bölgelerin geçilmesinde arama işlemine rehberlik etmek için birtakım kısıtlamaların oluşturulmasıyla ilgilidir. Bu kısıtlamalar, yasaklanmış olarak sınıflandırılan arama seçeneklerinin hariç tutulması ve değiştirilmiş değerlendirmelere ve seçim olasılıklarına dönüşüm gibi birkaç şekilde işler. Kısıtlamalar, hafıza yapılarından yararlanılarak oluşturulur ya da yaratılır. Bu kısıtlamalar hem “yasaklı” olarak arama işleminin direk dışında kalarak, hem de değiştirilmiş değerleri ve aramanın olasılığını çevirerek çeşitli şekillerde çalışır. Tabu aramanın çıkış önermesi, hafızanın kullanımını etkili kılmada kritik bir yol oynayan akıllıca yasaklamalardır (Glover ve Laguna, 1997).

Tabu arama, yerel arama yapan sezgisel algoritmaları kullanan bir meta-sezgisel işlemdir. Meta-sezgiseller yerel optimuma takılmayı önlemeyi sağlayarak arama uzayının kullanılmasına izin veren ileri düzey stratejilerden meydana gelir. Diğer tümleşik yaklaşımlarda olduğu gibi, TA optimuma yakın çözümlerin aralığını ya da optimum çözümleri bulmayı amaçlayan arama uzayındaki bir çok değişimi yerine getirir. “Tabu” isimlendirmesi, arama uzayının yakın zamanda incelenmiş belirli alanlarının yeniden incelenmesini engellemek için algoritmanın bu alanları tabuya çevirmesi (ya da engellemesi) ile ilgilidir. Arama, belirli bir zaman periyodu için (tabu süresi) tabu olarak belirlenmiş bölgelere ait çözüm noktalarını tanımlayan özellikleri içeren alternatiflerin incelenmesini göz önünde bulunduracaktır (Monticelli vd., 2008).

TA'nın esasları çok önceleri verilmesine rağmen, bugün kullanılan şekli Glover tarafından açıklanmıştır. Tabu Arama kavramı ilk olarak Glover (1986) tarafından insan hafızasının çalışmasından esinlenerek ortaya atılmıştır ve Hansen (1986) tarafından geliştirilmiştir. Esas olarak başka bir sezgisel yöntemin üzerine kurulan meta-sezgisel bir yöntem olarak düşünülebilir. Tabu Arama, kısıt metotları ve düzlem kesme yaklaşımlarından ortaya çıkmıştır. Tabu Arama, uygunluk ve yerel optimal sınırlardan geçen metotlara dayanan köklere sahiptir (Değertekin vd., 2006).

Glover ve Laguna (1997) Tabu Aramanın amacını şu şekilde belirtmiştir; kendi beynimizde cisimlenen organizasyon şekillerinin belirsiz işaretleri olan fiziksel ve biyolojik işlemler akıllı davranışlar içerebilir. Büyük ihtimalle evrim, genellikle çevremizde kendini gösteren problemleri çözme yeteneklerimizi geliştirir. Teknolojinin sonucu olarak oluşan problemleri içeren durumlarla karşılaştırıldığında problem çözme yeteneğimiz daha az etkileyici olabilir. TA araştırmasının asıl amacı hafıza ve strateji bileşimlerini tanımlamaktır. Eğer bu arayış zekanın farklı şekillerini anlamayı sağlayabilirse, bu hoş bir sürpriz olur. Bu açıdan, hafıza tabanlı arama metotları kavramsal davranışlarla karşılaştırılmak için faydalı bir şeye sahip olabilir. Şimdiye kadar, arama ve psikolojinin tamamlayıcı alanları birbirlerinden büyük ölçüde ayrılmıştır. Bu durum, uyumsal hafıza işlemi ve sistemleştirilmiş gelişmiş problem çözme arasında ilişkiler bularak değişebilir.

TA algoritmasında belirli kısıtlar altında bir $f(x)$ amaç fonksiyonunun optimal (minimum ya da maksimum) değeri araştırılacaktır. Tanımlamalar yapılırken ilgilenilen problem, $x \in X$ kısıtı altında $f(x)$ amaç fonksiyonunun minimum yapılması olarak düşünülmüştür. Optimal değeri araştırılan $f(x)$ amaç fonksiyonu doğrusal olabilir ya da olmayabilir. $x \in X$ koşulu, x vektörü üzerindeki kısıtları göstermektedir.

Tabu arama öncelikle, akıllı olarak nitelendirilmek için hassas bir araştırmayı ve uyumsal hafızayı kapsayan problem çözümüne dayanır. Basit Tepe Tırmanma (BTT), iyi bir benzetmedir; tırmanıcı geçici olarak geçtiği yolun anahtar noktalarını hatırlar (uyumsal hafızayı kullanma) ve yol boyunca stratejik seçimler yapabilmelidir (hassas araştırma). Tabu aramanın uyumsal hafıza özelliği (mevcut alternatiflerin daha önceki benzer bölgelerin tırmanışı ile ilişkisini analiz edebilen tepe tırmanıcının kıyaslaması ile önemi belirtilir), çözüm uzayını arama işleminin verimli ve ekonomik bir şekilde yerine getirilmesine izin verir. Yerel seçimlere arama sırasında derlenen bilgiler rehberlik ettiği için TA ağırlıklı olarak örneklemenin bir şeklini geliştiren yarı-rassal işleme bağlı hafızasız yapıyla ters düşer. Hafızasız metotların örnekleri yarı-doyumsuz sezgiseller, geleneksel tavlama ve evrimsel yaklaşımları içerir. Uyumsal hafıza dal ve sınır stratejilerinin klasik tasarımı olan sabit hafızaya ters düşer (Glover ve Laguna, 1997). BTT yönteminde algoritma yerel optimumda takılır ve bir kısır döngü oluşur TA

bu kısır döngüden kurtulmak için hafıza yapısını kullanır. Hafızanın kullanımı, şimdiki çözümün komşusunun bazı alt kümelerinin seçimi ile bir sonraki çözümü sınırlandırır.

Genetik Algoritma ve Benzetimli Tavlama ile karşılaştırıldığında Tabu Arama daha atılğan bir yoldaki çözüm uzayını araştırır. Tabu Arama algoritmaları mevcut çözüm olan bir durumla başlar. Algoritmanın her tekrarında mevcut durum için bir komşuluk yapısı tanımlanır; bir hareket, bu komşuluktaki en iyi çözüme gitmektir (örneğin bir minimizasyon probleminde algoritma en az maliyeti sunan duruma gider). Genellikle sadece en ümit verici komşular geliştirilir, aksi takdirde problem kolay kontrol edilemeyen bir hale gelebilir. Yerel aramalar için gradiyent tipi algoritmaların aksine tabu aramada komşuluk dinamik olarak güncellenir. Diğer bir farkı ise, daha yüksek maliyetli çözümlere izin vermesidir (bu yerel minimum noktalardan kurtulmayı sağlayan yöntemdir). Tabu Arama algoritmasının gerekli bir özelliği, geçici süreliğine yasaklı (tabu) olarak sınıflandırılan arama alternatiflerinin tamamen dışında tutmasıdır. Bu nedenle, hafızanın kullanımı bu algoritmalarda can alıcı rol oynar: tabu kısıtlamalarını aklında tutar. Hassas inceleme akıllı aramanın temel ilkelerini birleştirir (yeni cazip bölgeleri araştırırken iyi çözüm özelliklerinden faydalanılır). Tabu arama uyumsal hafıza ve hassas inceleme ile alakalı olan mekanizmaların avantajına sahip yeni ve daha etkili yollar bulmayla ilgilenir. Yeni tasarımların ve stratejik bileşimlerin gelişimi TA'yı deneysel çalışma ve araştırma için üretken bir alan yapar. Tabu arama, aslında yapay zekada kullanılan fikirlerden geliştirilmiştir. Genetik algoritma ve benzetimli tavlama gibi diğer tümleşik yaklaşımların aksine tabu aramanın kökeni biyolojik ya da fiziksel işlemlerle ilgili değildir. Tabu arama Glover tarafından 1980'lerin başlarında tasarlandığından beri fen ve mühendislikteki bir çok karmaşık probleme başarıyla uygulanmaktadır. Bunlardan, elektrik enerjisi şebekesi problemlerindeki uygulamaları önemli ve gelişmekte olan çalışmalardır. Uzun dönem aktarım şebekesi genişleme problemi ve ana şebeke hattındaki optimum kapasite yerleştirme problemi gibi dağıtım planlama problemleri gibi örnekleri içerir (Glover ve Laguna, 1997).

Tabu Aramada arama boyunca en iyi komşu çözümlerin elde edilmesini sağlayan hareketler yapay bir hafızaya kaydedilir. Bu hafıza kısa dönemli hafıza olarak adlandırılır. Bu yapay hafıza kullanılarak aramada önceki çözümlerin tekrar elde

edilmesini sağlayacak hareketler yasaklanır. Arama esnasında yapılmasına izin verilmeyen bu hareketlere tabu denir. Tabu hareketler tabu listesi adı verilen ve yapay hafıza görevi gören tek boyutlu bir listede muhafaza edilir. Tabu listesine kaydedilen her hareket bir çözüm değişkenini temsil eder (Değertekin vd., 2006).



Şekil 3.1 : Temel Tabu Arama Algoritması

Tabu Aramada her çözüm değişkeni için tanımlanan komşu çözümler içerisinde en iyi olan çözüm tabu listesi de kontrol edilerek mevcut çözümün yerini alır. Arama işlemine mevcut çözümün yerini alan en iyi komşu çözüm ile devam edilir. Aynı işlem diğer çözüm değişkenleri için de tekrarlanarak bir iterasyon tamamlanır. Kabul edilen durdurma kriteri sağlanıncaya kadar her iterasyon için bu işlemler tekrarlanır.

Tabu aramanın diğer mekanizmaları “yoğunlaştırma (kuvvetlendirme) ve çeşitlendirme”dir: yoğunlaştırma mekanizması ile algoritma yerel optimum noktaya götürebilen cazip alanların daha kapsamlı bir araştırmasını yapar; diğer bir yandan çeşitlendirme mekanizması ile, arama daha önce ziyaret edilmemiş bölgelere taşınır, yerel minimum noktalardan kaçınmak için önemli bir şeydir. Tabu arama, akıllı bir yöntemdeki karmaşık bir problemi çözmek için bütünleşik yolda uygulanmış prensipler (ya da fonksiyonlar) kümesinden oluşur. Glover’a göre: Tabu arama, akıllı olarak tanımlanmak için uyarlamalı hafıza ve esnek araştırmayı kapsayan problem çözümü önerisine dayandırılır. Uyarlamalı hafızanın kullanımı fizik ve biyolojinin benzetmelerinden esinlenmiş hafızasız tasarıma ve sınıf, vb. ile örnek verilmiş sabit hafıza tasarımına ters düşer. Tabu aramada esnek aramayı vurgulama, belirleyici ya da olası bir uyarlamada olup olmadığı, kötü bir stratejik seçimin iyi bir rastgele seçimden daha fazla bilgi sağladığı varsayımından çıkartılır (Monticelli vd., 2008). Eğer sadece

iyileştirmelere bakılırsa bu tam doğru bir Tabu Araması olmaz. TA şimdiye kadar elektronik devre tasarımı, iş çizgeleme, kesim problemleri, telekomünikasyon ağları, düzlem ve uzay kafes sistemlerin optimizasyonu gibi birçok farklı alanda uygulanmıştır (Glover ve Laguna, 1997).

3.2.1. Komşu Arama Algoritması

Tabu aramanın kritik noktası, gelişmenin komşuluk yapısına ihtiyacının olmasıdır. Tabu Arama çözüm uzayındaki arama işlemini komşu arama ile gerçekleştirmektedir. Komşu aramada, herhangi bir olası çözümün diğer olası çözümlerden oluşan komşuları olduğu kabul edilir. Bir komşuluktaki durumların sayısı çok fazla veya bu durumların kalitesi çok çeşitli olabilir. Optimizasyon probleminde ise olası birçok çözüm içinde amaç fonksiyonunu minimum yapan çözüm aranır. Bu amaçla çözüm uzayında hareket mekanizmasıyla mevcut çözümden bu çözüme komşu olan çözümlere gidilir. Komşu aramada komşuluğun karşılıklı olduğu kabul edilir. Yani X çözümü, Y çözümünün bir komşusu ise, Y çözümü de X çözümünün bir komşusudur. Temel TA algoritmasının her tekrarında, mevcut çözümün komşularından tabu içermeyen ya da içeriyorsa da amaç kriterini sağlayan en iyi çözüm tanımlanıp hesaplanır. TA'nın temel stratejilerinden biri mevcut durumun komşularından en iyi olan duruma hareketi içerir. Genelde komşuluğun genişliği algoritma ile geliştirilenden daha geniştir. Böylece komşuluğun en cazip kısmıdır aslında. Tabu arama ile ilgili son çalışmalar komşuluk boyutunu azaltmak için genel amaçlı stratejiler öneriyor. Bu stratejilerin her problemde çalışması gerekmez ve probleme özel stratejiler geliştirilmelidir (Değertekin vd., 2006).

Bir hareket aslında bir komşu seçme fonksiyonudur ve $j=N(i)$ ile gösterilir. Bu notasyon i çözümünden j çözümüne hareket etmeyi ifade eder. Bunun tersi ise j çözümünden i çözümüne harekettir.

i veya x = mevcut çözüm

j veya x' = komşu çözüm

$N(i)$ veya $N(x)$ = mevcut çözümün tüm komşuları

$f(i)$ veya $f(x)$ = mevcut çözümün maliyeti

$f(j)$ veya $f(x')$ = komşu çözümün maliyeti

ise $f(j) < f(i)$ veya $f(x') < f(x)$ koşulunu sağlayan çözümler aranır.

Orjinal Komşu Arama Algoritması mevcut çözümün tüm komşuları $N(i)$ araştırıldığında sonlanmaktadır. Ancak böyle bir yaklaşım hareketler değişken olduğundan mümkün olmayabilir. Bu nedenle bitirme koşulu belirlenir. Bitirme koşulu belirli bir çevrim sayısı t olarak verilebilir.

Tabu süresi (ϵ), tüm hareket sayısı olan t 'den küçük ($\epsilon < t$) olmalıdır. Tabu süresinin bir anlamının olabilmesi için $\epsilon < t-1$ olarak belirlenmelidir.

Örnek bir çözüm (Glover, Laguna, 2002),

2	5	7	3	4	6	1
---	---	---	---	---	---	---

Bu çözümün 21 (7 eleman, 2'li değişim) komşuluğu vardır.

Örnek komşu çözümler,

2	6	7	3	4	5	1
---	---	---	---	---	---	---

1	5	7	3	4	6	2
---	---	---	---	---	---	---

Bir hareketin niteliği: x den x' çözümüne bir hareket sonucu çözümün değişen herhangi bir özelliği ya da bileşeni bu hareketin bir niteliği olabilir. Tek bir hareket birden fazla niteliğe de sahip olabilir (Sait ve Youssef, 1999).

3.2.2. Temel Tabu Arama Algoritması

Kısa dönem hafızalı bir tabu arama algoritması şu özelliklere sahiptir (Monticelli vd., 2008):

- Uygun ve uygun olmayan çözümleri içeren arama uzayında k_T hareketlerin işlemidir. k_T hem daha önceden tanımlanmıştır hem de saptanabilir.
- Mevcut durumun komşuluğu minimum maliyet için aranmıştır. Eğer durum tabu değilse ya da tabuysa ama amaç kriterini sağlıyorsa onaylanır. Mevcut çözümden daha yüksek maliyetli durumlara hareketlere izin verilir.
- Her tekrarda tabu özelliği listesi güncellenir.

3.2.3. Problem Formülasyonu

Tanımlanan $f(x)$ maliyet veya kar fonksiyonu, lineer veya nonlineer olabilir. X kümesi, x vektörü üzerindeki sınırlamaları ifade etmektedir. Tabu arama, sıradan bir lokal araştırma veya komşuluk araştırma algoritması gibi çalışmaya başlar. Her bir $x \in X$, $N(x)$ olarak gösterilen bir komşular setine sahiptir. x 'in komşu setine alınan komşuları da x üzerinde tanımlanan kısıtlamalara uyar. x çözümünden bir $x' \in N(x)$ çözümüne gitmek için yapılan bir işlem "hareket" olarak tanımlanır (Monticelli vd., 2008).

Tabu Aramada herhangi bir iterasyonda x çözümünden bu çözümün en iyi komşusu olan x' çözümüne bir hareket gerçekleştiğinde bu hareket tabu listesine kaydedilir. Mevcut bir x çözümünden h_1 hareketi ile en iyi x' komşu çözümüne gidilsin, $x'=h_1(x)$. Bir sonraki iterasyonda mevcut çözüm olan x' çözümünün en iyi komşu çözümü x olabilir. x' çözümünden h_2 hareketi ile x çözümüne gidilir, $x=h_2(h_1(x))$. Bu olasılığın gerçekleşmesi durumunda aramada aynı çözüm etrafında bir çevrim oluşacaktır. Tabu listesi ile bu durum yasaklanmakta ve aramanın çözüm uzayında farklı alanlara yönelmesi sağlanmaktadır (Değertekin vd., 2006).

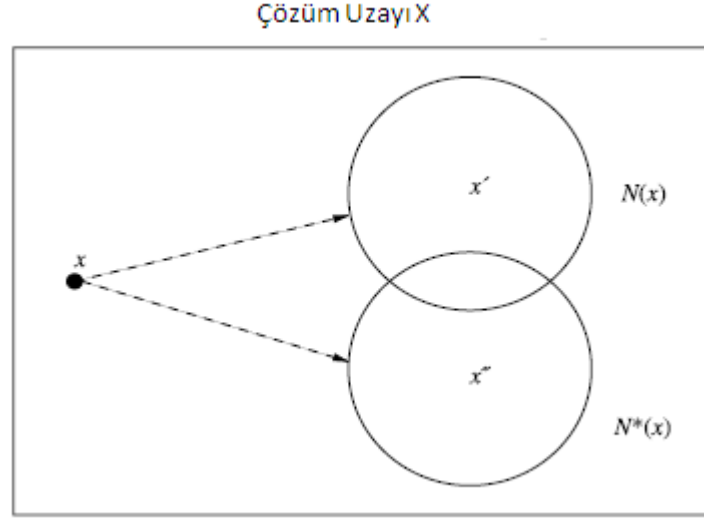
Genellikle TA algoritmaları problem çözümünde şu şekilde formüle edilir:

Amaç: $f(x)$ fonksiyonunu optimize etmek

Kısıtlamalar: $x \in X$

Eğer x bir düzense (ya da bir karar değişkeni), $f(x)$ amaç fonksiyonu ve X de arama uzayıdır. Ayrıca $f(x)$ 'in ve X 'in dışbükeyliği ya da $f(x)$ 'in diferansiyellenebilirliği hakkında hiçbir varsayım yapılmadığı belirtilmiştir. Bir tür tümleşik optimizasyon problemleri yukarıdaki gibi bir cebirsel kısıtlamalar kümesine bağlı olarak amaç fonksiyonunun minimizasyonu olarak gösterilebilir. Kısıtlamaların basitçe cebirsel kısıtlamalar olarak gösterilemediği durumlar da vardır. Belirli matematiksel yaklaşımlar için rahatsız eden bu kısıtlamaların normalde direk olarak cebirsel kısıtlamalarda kullanılmayan TA kolayca üstesinden gelir. Tabu arama bir problemi öncelikle verilen bir x durumunda bir yerel sezgisel arama uygulayarak çözer, x 'in komşulukları, $x' \in N(x)$ 'in tüm durumlarının kümesi olarak tanımlanır. x' x 'in komşusu olabilmesi için x 'in komşuluğunun yapısı tanımlanmalıdır. Yerel arama algoritmaları amaç

fonksiyonundaki en geniş azalma miktarını gösteren x' durumunu yönlendiren değişimi bulur. Bu işlemin tekrarı en sonunda yerel optimal sonuca yönlendirir. Tabu arama basit yerel arama algoritmasından farklıdır (Şekil 3.1) (Monticelli vd., 2008):

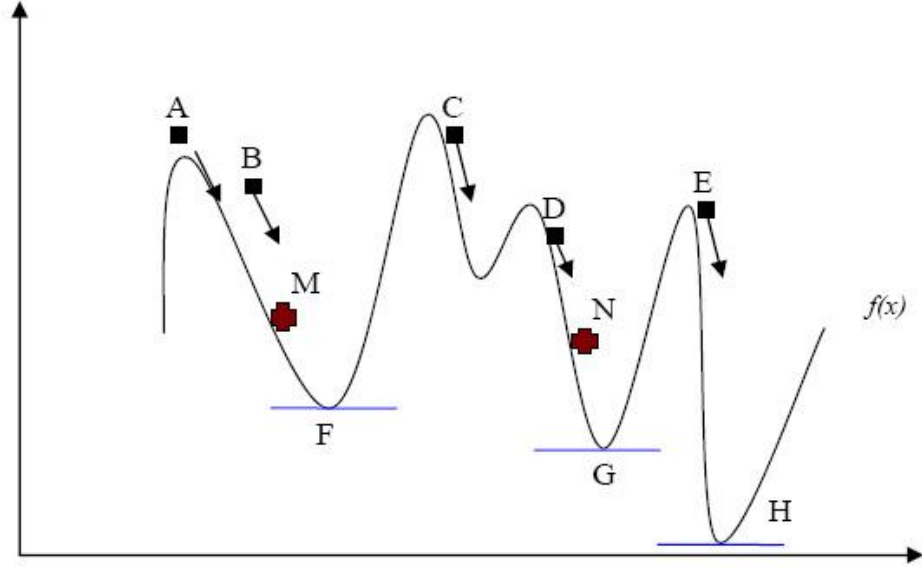


Şekil 3.2 : Tabu Aramada mevcut çözümden komşu çözümlere geçiş (Monticelli vd., 2008)

Mevcut çözümün sahip olduğundan daha iyi bir amaç fonksiyonuna sahip durumlara yönlendiren değişime izin verilir. x' 'in komşuluğu $N(x)$ sabit değildir, değişebilir. Değiştirilmiş komşu $N^*(x)$ farklı yollardan saptanır. Yasaklanmış durumların özelliklerini içeren tabu listesi kullanılır. Bu durumda $N^*(x) \subset N(x)$ yukarıda belirtildiği gibi döngüden ve yerel optimumdan kurtulmak için faydalıdır. Yerel aramayı hızlandırmak için komşuların miktarını azaltacak stratejiler kullanılır. Önceki durumda olduğu gibi $N^*(x) \subset N(x)$ komşuları azaltır. Sözde seçkin durumları kullanmak. Bu durumda $N^*(x) \subset N(x)$ doğruluğu gerekli değildir. $N(x)$ 'in optimizasyon işlemi esnasında yeniden tanımlanması; normalde problemin özel niteliklerinden faydalanmak için yapılır.

Şekil 3.2'de iki boyutlu bir uzayda amaç fonksiyonu eğrisi ve rasgele belirlenmiş $\{A, B, C, D, E, F, G, H, M, N\}$ noktaları gösterilmektedir. Eğer başlangıç noktaları A, B, C, M, N noktalarından herhangi birisi olursa algoritma en iyimser ihtimalle yerel en küçük noktaları olan F ve G noktalarında duracaktır. Bazı durumlarda iterasyon miktarı

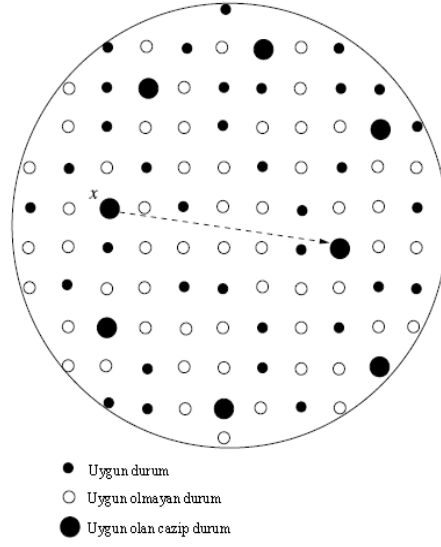
nedeniyle belki de A ve B’de başlayan M noktasında D’de başlayan ise N noktasında durabilir (Ünler,2006).



Şekil 3.3 : Örnek yakınsama grafiği (Ünler, 2006)

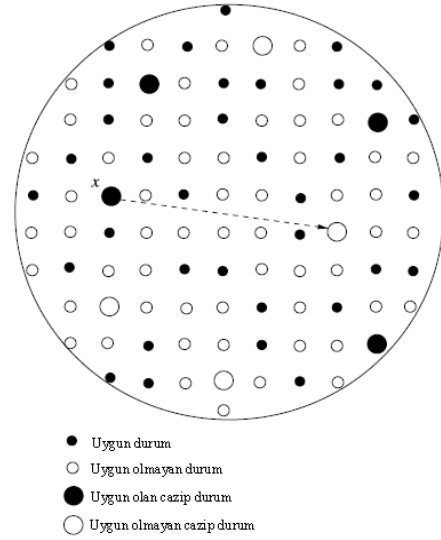
Halbuki Şekil.3.3’teki amaç fonksiyonuna göre tümel en iyi nokta H noktasıdır. Basit Tepe Tırmanma Algoritmasını kullanan yöntemler bu tür problemlerde genellikle ilk gördükleri yerel en iyi noktaya takılmaktadır. Birden fazla yerel en küçük noktası olan problemler ancak sezgisel yöntemler yardımıyla çözülebilmekte ve tümel en iyi olmasa bile buna yakın çözümlere ulaşılabilir. En iyileme yazını incelendiğinde bu tür problemlerin çözümüne yönelik olarak geliştirilmiş bir çok sezgisel yöntem bulunmaktadır (Ünler, 2006).

Şekil 3.4’te üretken x noktası komşuluğu ile gösterilmiştir. Verilen bir değişim mekanizması, x ’in bir komşusu x ’ten yapılacak tek değişimle ulaşılabilir her hangi bir noktadır. Hem uygun hem de uygun olmayan komşuluklar vardır. Uygun olanlar arasından sadece cazip durumlarla ilgilenilir. Monticelli vd. (2008) bu durumu şu şekilde açıklamışlardır:



Şekil 3.4 : Mevcut bir durumun komşuluğu (Monticelli vd., 2008)

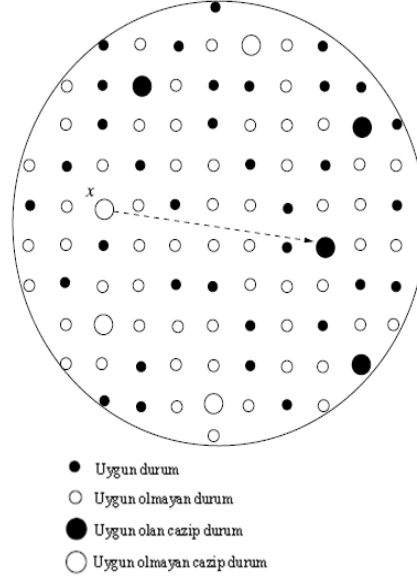
Şekil 3.4 x 'in uygun olmamasına rağmen ilgilenilen komşularını içerir. Belirli koşullarda uygun olmayan cazip durumlara doğru geçici değişimlerin sebebi, uygun cazip çözümlere ulaşmaya yönlendiren en kısa yol olabilmesidir.



Şekil 3.5 : Uygun bir durumdan uygun olmayan bir duruma geçiş (Monticelli vd., 2008)

Şekil 3.5 durumu uygun olmayan cazip noktadan uygun olan cazip noktaya taşıyan tümleyen değişimleri gösterir. Bu tür strateji yerel optimal çözümlerin çok olduğu kompleks sistemlerin iyi bir örneği olan ulaştırma geliştirme planlama probleminde başarıyla kullanılmaktadır. Uygun olmayan noktalara götüren değişimler uygunsuzluğun maliyetini gösteren amaç fonksiyonunda cezalı bölgeleri kapsamıyla

kolaylaştırılmıştır; böylece, mevcut maliyetteki azalım uygunsuzluğa bağlı kısımların maliyeti için dengelenecekten daha fazla ise, değişime izin verilir.



Şekil 3.6 : Uygun olmayan bir durumdan uygun olan bir duruma geçiş (Monticelli vd., 2008)

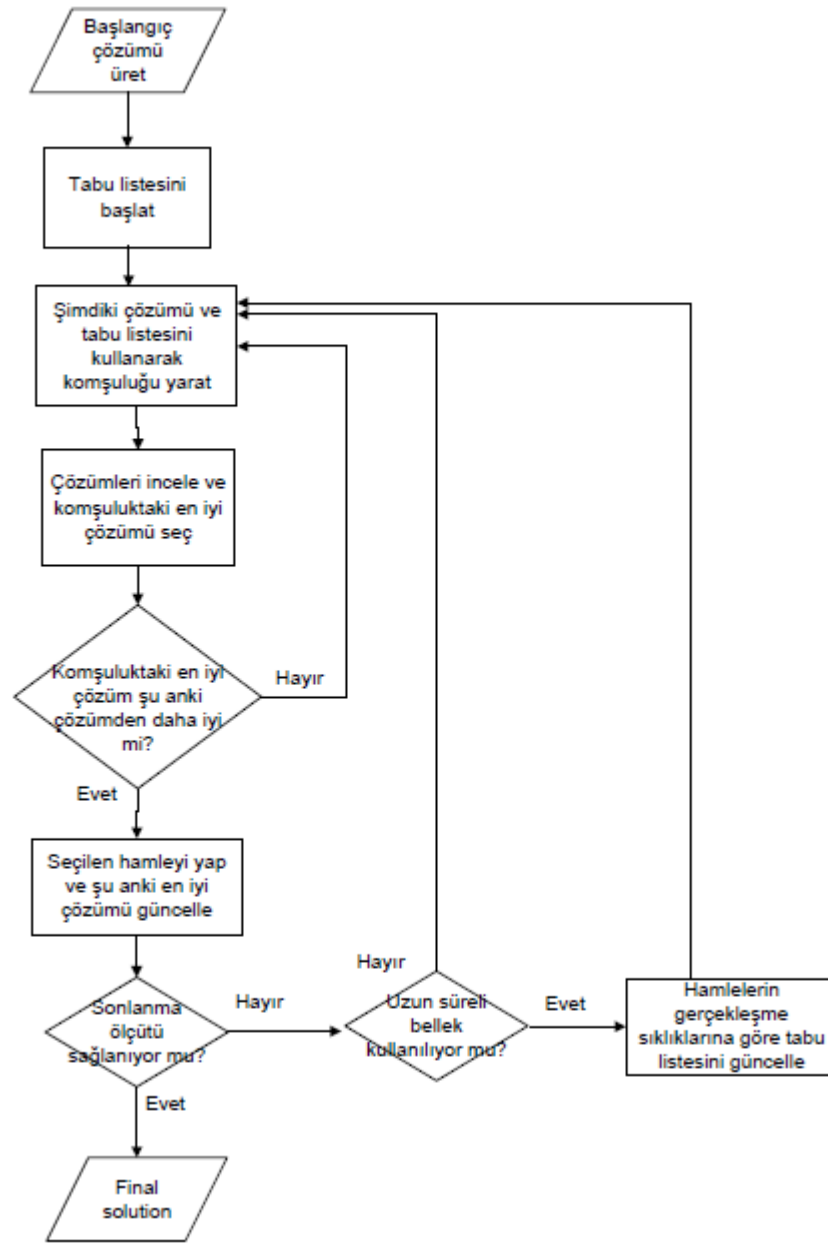
3.2.4. Tabu Aramanın Temel Kavramları

Başlangıç çözümünün oluşturulması: Genelde başlangıç çözümü rassal olarak elde edilir. Ancak ilgilenilen problem için geliştirilmiş olan sezgisel bir algoritmadan yararlanarak elde edilmesi de mümkündür (Ünler, 2006).

Hareket mekanizması: Tabu Arama aslında dinamik bir komşuluk arama metodudur. Mevcut çözüm x ile temsil edilsin, X mümkün çözümler kümesi olsun. x 'in herhangi bir işlem görmesiyle y çözümü bulunuyorsa, “ y çözümü x 'in komşusudur” denilir, x 'in işlem görmesi ise “hareket” olarak tanımlanır (Ünler, 2006).

Hareket değeri: Bir hareketin değeri amaç fonksiyonundaki azalıştır .

Komşu çözüm: Bir hareket mekanizması kullanılarak x mevcut çözümünden oluşturulan diğer çözüme komşu çözüm denilmektedir. Komşu çözüm x' ile gösterilmektedir. $\forall x \in X$ olmak üzere, $N(x) \subset X$, x' 'in komşuluk setini ifade etmektedir. Yerel arama için önemli bir kavram olan komşuluk yapısı, yerel arama tabanlı tabu arama algoritması için de en önemli faktörlerden birisidir.



Şekil 3.7 : Tabu Arama Algoritmasının Temel Akış Şeması (Gözüpek ve Genç, 2009)

Aday liste stratejileri: Gerçekleştirilen hareketlerin oluşturdukları komşulukların alt kümelerini içeren listedir. Kombinatoriyal optimizasyon problemlerinin çözümünde karşılaşılan ortak bir nokta, herhangi bir x çözümü için oluşturulabilecek $N(x)$ komşuluğunun oldukça büyük olmasıdır. Özellikle problem boyutu büyük olduğunda $N(x)$ komşuluğundaki tüm seçeneklerin incelenmesi hesaplama zorluğu nedeniyle çok pahalı hatta imkânsız olabilir. Böyle durumlarda, incelenen çözümlerin sayısını aday liste stratejileri kullanımıyla kısıtlamak mümkündür (Glover, 1989a).

Monticelli vd. (2008) arama işlemini temel olarak şu şekilde açıklamıştır; ilk olarak komşuluk tanımlanır, geliştirilir. Çünkü normalde hem komşuluk çok geniştir hem de her alternatifin gelişimi zaman alır, aramadaki en cazip komşuları sınırlandıracak bir eleme işlemi uygulanmalıdır. Güç sistemi uygulamalarında, örneğin, gelişim hem lineer problemlerin, hem lineer olmayan problemlerin, hem de güç akış problemlerinin çözümü için gereksinimleri ifade etmelidir. Literatürde komşuluğun eleme için dört strateji kullanılmaktadır:

- 1) amaç ilavesi,
- 2) seçilmiş aday liste,
- 3) başarılı filtreleme stratejisi,
- 4) ardışık yayılan aday listesi

Amaç ilavesi tekniği yeni aday çözümler için göz önünde bulundurulmuş bir durum için minimum kalite düzeyi tanımlanır. Bu metod muhtemel komşuları eşik değeri geçene kadar belirtilmiş listeye koymayı ve analiz etmeyi gerektirir. Eşik değere ulaşıldıktan sonra çok az bir durum değişimi olur ve işlem sona erer. Bir azalıştan ya da aşırı derecede fazla miktarda komşuyu analiz etmekten kaçınmak için daha alt ve daha üst sınırlar gerekir. Eşik değeri tanımlamak için son iki alternatif söz konusudur :

- 1) son k tane harekette ziyaret edilen en iyi durumun amaç fonksiyonunun değeri
- 2) mevcut çözümün amaç fonksiyonunun değeri.

Seçilmiş aday liste tekniği, başlangıç durumunun komşuluğunun en iyi n_p elemanı ile şekillenen bir esas liste ile başlar. Sonra ilk n_p komşu kümesi değişmeden aynı kalırken bir dizi hareket gerçekleştirilir. Hem verilen amacı sağlayan bir durum bulunduğunda hem de maksimum hareket sayısı sağlandığında yeni bir esas liste yapılır ve işlem tekrar edilir.

Başarılı filtreleme stratejisi tekniği normalde komşuluk yapısı değiş tokuş özellikleri ile tanımlandığında kullanılır. Bu, bir devirin eklenmesi ve diğer bir devirin çıkarılması (aday devirlerin değiş tokuşu) ile bir komşu elde edilmesidir, örneğin, ağ değişim planlaması. Azalan bir komşuluk iki kısa listenin tanımlanması ile elde edilir, biri eklenebilen elemanlarla, diğeri mevcut durumdan çıkarılan elemanlarla olur.

Ardışık yayılan aday listesi tekniği genetik algorithmada kullanılan popülasyonun konseptine benzer. Verilen başlangıç çözümü, bütün komşuluk geliştirilir ve en iyi komşuların listesi n_p 'de bir azalma oluşur. Bu durumlar mevcut durumlar (popülasyon) olarak adlandırılır. Daha sonra, her mevcut durumun komşuluğunun sayısında bir azalma geliştirilir ve her durum için yerine geçecek bir tane bulunur.

Tabu kısıtlaması: Hareketlerin seçilmiş niteliklerini tabu yaparak (yasaklayarak) önceden ziyaret edilmiş çözümlere geri dönmekten kaçınmak için yaratılan yasaklamalardır. Aramanın yerel optimal içeren bir bölgede sıkışıp kalmasını engelleyerek, böyle bölgelerin aşılmasına olanak sağlar (Aladağ, 2009).

Tabu listesi: Algoritma yerel minimuma hareket ederek başlar. Daha önce yapılmış hareketlere dönüş yapmayı engellemek için yöntem bir ya da daha fazla tabu listesi tutar. Daha önce ziyaret edilmiş veya herhangi bir nedenle ziyaret edilmesi istenmeyen çözümler Tabu Listesi adı verilen kısa dönem hafızaya benzer bir yapıda tutulur. Listenin amacı, önceden yapılmış bir hareketin tekrarından çok tersine dönmesini önlemektir. Tabu Listesi;

- Kronolojik bir yapıya sahiptir.
- Tabu arama belleğini biçimlendirir.

Tabu Aramada yapılan bir hareketin ne kadar süreyle tabu listesinde kalacağı tabu listesi uzunluğu ile belirlenir. Tabu listesinin yeniliğe dayanan bir hafıza yapısı bulunmaktadır. Buna göre tabu olan hareketlerin sayısı tabu listesi uzunluğuna eriştiğinde tabu listesinin en başında olan en eski hareket listeden çıkarılır. Liste bir sıra yukarı hareket ettirilerek listenin ikinci sırasındaki hareket birinci sıraya, üçüncü sırasındaki hareket ikinci sıraya gelir bu şekilde devam edilerek en son tabu olan hareket sondan bir önceki sıraya gelir. Boşalan son sıraya ise yeni yapılan hareket yerleşir. Yeniliğe dayanan hafıza yapısını kullanan Tabu Aramada tabu listesinin uzunluğu t oldukça önemlidir. t adet yeni hareket yapıldıktan sonra hareketin tabu durumu kaldırılır ve bu hareket tabu listesinden çıkartılır. Eğer tabu listesi çok uzun olursa aramada hareketlerin çoğu tabu olacağından Tabu Aramanın hareket kabiliyeti azalır. Tersine tabu listesinin çok kısa seçilmesi

durumunda ise aynı çözümler etrafında çevrim olabileceğinden arama başlangıç çözümünün yakınında lokal bir optimuma yakınsayabilir (Monticell, vd., 2008).

Elit çözüm: Yüksek kaliteli yerel optimal çözüm, elit çözüm olarak adlandırılır (Glover ve Laguna, 1997).

Hafıza: Tabu Aramanın en temel elemanlarından birisi hafızasıdır. Arama boyunca ortaya çıkan durumlar hafızaya kaydedilir. Bu hafıza kısa dönemli hafıza olarak adlandırılır. Yapılmasına izin verilmeyen hareketler “tabu” olarak adlandırılır ve esnek hafıza içinde “tabu listesi” adı altında kaydedilir. Bu hareketler belli bir süre sonra tabu listesinden çıkarılır ve yapılmasına izin verilir (Ünler, 2006). Tabu aramada hafıza yapıları dört boyuta dayanarak işlerler. Bunlar, *yakın geçmiş* (recency), *frekans* (frequency), *kalite* (quality) ve *etki* (influence) boyutlarıdır .

Tabu listesine dahil edilecek olan çözümlerin belirlenmesi bazı tabu şartlarına (tabu conditions) göre yapılmaktadır ve bu şartlar iki önemli faktöre dayalıdır. Bunlar, yakınlık hafızası ve sıklık hafızasıdır. Yakınlık ve sıklık hafızaları sırasıyla bir çözümün en son ne zaman ve hangi sıklıkla denendiğine ilişkin bilgileri saklamaktadırlar. Bu bilgiler, problemin yapısına göre oluşturulan bazı şartlar doğrultusunda değerlendirilir ve ilgili çözümün tabu listesine alınıp alınmayacağına karar verilmektedir. Böylece, kullanılan hafıza kriterleri yardımıyla, araştırma sırasında çözüm vektörüne ait elemanların belirli bir disipline bağlı olarak değişim göstermeleri sağlanır (Bağış, 2009).

Tabu arama metodu, hafızada tuttuğu bilgileri kullanarak araştırma işlemine yön verdiği için, akıllı bir metoddur. Tabu arama, geçmişte karşılaştığı durumları aklında tutar ve stratejik seçimler yaparak ilerlemeye çalışır. Glover ve Laguna, kötü bir stratejik seçimin iyi bir rastgele çözümden daha çok bilgi verebileceğini; hafızanın kullanıldığı bir sistemde belirli bir stratejiye göre yapılan kötü bir seçimin, stratejinin daha verimli hale getirilmesi için ipuçları içerebileceğini belirtmektedirler.

Tabu arama metodunda hafıza kullanımı iki şekilde ortaya çıkar: kısa dönem hafıza kullanımı ve uzun dönem hafıza kullanımı. Kullanılan hafıza tipi, mevcut komşuluktan

daraltılmış komşuluk, $N^*(x)$, elde edilirken fark edilir. Kısa dönem hafıza kullanımında $N^*(x)$, $N(x)$. in alt kümesiyken uzun dönem hafıza kullanımında $N^*(x)$ içerisinde $N(x)$. te bulunmayan çözümler olabilir. Tabu aramada x . in komşuluğu statik bir set değildir, aksine bu set araştırmanın seyrine göre değişir. x çözümü değişikçe $N(x)$ de değişir. Kısa dönem hafıza kullanımına örnek olan yakınlık tabanlı hafıza kullanımı ile, son yapılan hareketler tutularak bu hareketlerin yakın zamanda yapılması engellenir. Böylece araştırmanın lokal bir alanda dolanması önlenmeye çalışılır. Uzun dönem hafıza kullanımına örnek olarak sıklık tabanlı hafıza kullanımı verilebilir. Bu hafıza ile, araştırmanın başlangıcından itibaren araştırma ile ilgili bilgiler tutulduğundan, elde edilen bilgi daha kapsamlıdır. İlerleyen bölümlerde bu hafızaların kullanımı anlatılmaktadır (Monticell, vd., 2008).

Niteliksel (attributive) hafıza: Niteliksel hafıza bir çözümden diğerine yapılan harekette değişen çözüm nitelikleri hakkındaki bilgiyi kaydeder (Glover, 1995).

Kalite: Hafıza yapılarında kalite boyutu arama sırasında uğranmış çözümlerin derecelerini ayırma yeteneğine karşılık gelir. İşlemsel olarak kalite, iyi çözümlere götüren hareketleri desteklemek için ödüllerin, zayıf çözümlere götüren hareketleri engellemek için cezaların sağlandığı, yoğunlaşma stratejisine dayalı öğrenme için bir temel oluşturur (Aladağ, 2009).

Etki: Etki boyutu sadece kalite üzerinde değil, aynı zamanda arama sırasında yapılan seçimlerin etkileri üzerinde de durur. Bir açıdan kalite etkinin özel bir şeklidir. Kalite boyutunda ilgilenilen nokta, çözümlerin derecelendirilmesiyle, etki boyutunda önemli olan, çözümler arasındaki farklılığın büyüklüğüdür (Glover ve Laguna, 1997).

Tabu süresi: Belirli bir tabu niteliği barındıran bir hareketin ya da çözümün yasaklı kalma süresidir. Yöntem bu tabu listesindeki hamlelerin belirli bir süre yapılmasını yasaklar. Böylece arama yerel en iyi noktadan kurtularak asıl sonuca yakınsayabilir. Tabu Arama, önceden incelenmiş olan bir yol olmadığı sürece, her çözümü araştırabilen bir süreçtir. Buna göre tabu listesi uzunluğu t olmak üzere tabu listesine kaydedilen bir hareketin, bu hareketten sonra t adet yeni hareket yapıncaya kadar tekrar edilmesine

izin verilmez. t adet yeni hareket yapıldıktan sonra hareketin tabu durumu kaldırılır ve bu hareket tabu listesinden çıkartılır.

Yakınlık tabanlı hafıza, tabu aktif elemanları tutan bir tabu listesi ile çalışır. Her tabu aktif eleman, belirli bir iterasyon süresince tabu listesinde tutulur. Bu süre tamamlanınca tabu aktif eleman listeden çıkarılır ve bu elemanları içeren çözümlerin tabu olma durumu ortadan kalkar. Böylece ilerleyen iterasyonlarda bu elemanları içeren çözümler ziyaret edilebilir. Tabu listesinde hangi bilgilerin tutulacağı ya da tabu listesinin yapısının hangi bilgileri tutacak şekilde tasarlanacağı, çözülmesi istenen probleme göre karar verilmesi gereken konulardır. Tabu listesine alınan bir elemanın ne kadar zaman bu listede kalacağı yani tabu tutulma süresi de probleminin boyutuna göre seçilmelidir. Tabu tutulma süresi için sabit bir değer verilebileceği gibi bu süre tabu tutulacak elemanın durumuna göre de belirlenebilir. Çok kısıtlayıcı nitelikteki elemanlara kısa tabu süreleri atanırken, tabu olma durumunun araştırma seyrini daha az etkileyen elemanlara daha yüksek tabu tutulma süresi verilebilir. Tabu tutulma süresine çok düşük bir değer verilmesi algoritmanın bir alanda takılmasını (cycling) engellemeyebilirken; bu sürenin yüksek tutulması çözümlerin kalitesinde bozulmaya neden olabilir (Glover ve Laguna, 1997).

Tabu süresi, bir özelliğin değişmeden cezalı olarak kaldığı tabu arama tekrarının sayısıdır (yani hareketlerin veya değişimlerin sayısıdır). Tipik bir uygulamada, her biri farklı süreli çoğu tabu listesi aynı anda korunabilir. Böylece, tek bir özellik değil özelliklerin kombinasyonları da yasaklanabilir. Çok nadiren, bir tabu istenmeyen bir durum hakkında tüm bilgileri verilerek belirtilebilir; normalde, sadece durumun belirli özellikleri tabu listesine konur. Bir tabu süresi hem statik (önceden tanımlanmamış) hem dinamik (önceden belirtilmiş) olabilir (Monticelli vd., 2008).

Bir tabu kısıtlamasını tanımlayan niteliklerin farklı türlerinin tabu sürelerinin belirlenmesinde farklı değerler kullanılmasına izin vermek genellikle akıllı bir tercihtir. Örneğin bazı nitelikler bir tabu kısıtlamasına diğerlerinden daha güçlü katkıda bulunabilirler. Bu durumda, kısıtın çok fazla katı olmasından kaçınmak için sözü edilen niteliklere daha kısa bir tabu süresi verilmelidir (Sait ve Youssef, 1999).

Tabu tutulma süresinin deęişken seçilmesi ile arařtırmada bir bölgenin detaylı bir şekilde incelenmesi ve ardından çözüm uzayının başka bölgelerine gidilmesi arasında bir denge kurulabilir. Dinamik tabu süresi belirlenmesi işlemi; rastgele dinamik ve sistematik dinamik olmak üzere iki şekilde gerçekleştirilebilir (Glover, 1989):

- Rastgele dinamik tabu süresi (random dynamic tabu tenure): Dinamik tabu süresi belirlenmesi iki şekilde yapılabilir. Her iki şekilde de t_{min} ve t_{max} arasında deęişen bir tabu tutulma süresi kullanılır. İlk durumda, seçilen tabu süresi belirli bir iterasyon sayısı kadar sabit kalır, bu süre sonunda tekrar bir tabu tutulma süresi seçilir. İkinci durumda ise her tabu olan eleman için sınırlar içinde olan, öncekinden farklı bir tabu tutulma süresi seçilir.
- Sistematik dinamik tabu süresi (systematic dynamic tabu tenure): Sistematik dinamik tabu süresi belirlemek için, içinde tabu tutulma sürelerinin tutulduęu, t_{min} ve t_{max} arasında deęişen elemanlardan oluşan bir liste tanımlanır. Bu listedeki tabu süreleri sırayla, baştan sona, sonra sondan başa gibi deęişik şekillerde kullanılır. Seçilen deęer belirli bir iterasyon sayısı kadar sabit tutulur, sonra sıradaki deęer kullanılır.

Durdurma kořulu: Tabu Arama algoritması bir veya birden fazla durdurma kořulunu sağlayıncaya kadar aramasını sürdürür. Bazı durdurma kořulları (Ünler, 2006);

- Seçilen son çözümün komşusunun olmaması
- Belirli bir iterasyon sayısına ulařılması
- Belirli bir çözüm deęerine ulařılması
- Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi

Tabu arama algoritması bir başlangıç çözümü üreterek aramaya başlar. Algoritmanın her bir yinelemede tabu olmayan bir hareket ile mevcut çözümün komşuları arasından bir tanesi seçilerek deęerlendirilir. Eđer amaç fonksiyonunun deęerinde bir iyileşme sağlanmışsa komşu çözüm mevcut çözüm olarak dikkate alınır. Seçilen bir hareket tabu olmasına rağmen tabu yıkma kriterlerini sağlıyorsa mevcut çözüm oluşturmak için kullanılabilir. Geriye dönüşleri önlemek için bir takım hareketler tabu

listesine kaydedilerek tekrar yapılması belli bir süre için yasaklanabilir. Belirlenen bir durdurma koşuluna göre algoritmanın çalışması sonlandırılır (Ünler, 2006).

Tabu Arama iyi çözümlerin başarısını devam ettiren ve iyi çözüm verecek yeni alternatifleri bulan hassas aramalar yapar. Tabu Aramanın sahip olduğu esnek hafıza sayesinde daha esnek bir arama davranışı ortaya çıkar. Bu sebeple hafıza temelli stratejiler Tabu Arama yaklaşımlarının ayırt edici özellikleridir. Bu noktada artık Tabu Aramanın ortaya attığı bellek yapısından yararlanır. Yakın geçmişte gerçekleşen bir hareketin tekrarlanmasına dayalı bellek yapısı, en sık kullanılan bellek yapısıdır ve tabu süresini bellekte tutar (Monticelli, vd., 2008).

3.2.5. Seçim Kurallarının Değiştirilmesi

Tabu arama algoritmasında seçim kuralları şu şekilde değiştirilir: Elde edilen geçmiş bilgi doğrultusunda bir hareketin veya o hareketin üreteceği bir çözümün değeri, $ÇözümDeğeri^* = ÇözümDeğeri + d \times Ceza$ şeklinde verilen ceza fonksiyonu kullanılarak değiştirilir. Burada $ÇözümDeğeri$, örneğin ilgilenilen hareketin ürettiği çözümün amaç fonksiyonu değerini gösterebilir. $Ceza$ değerinin genellikle frekans ölçütlerinin bir fonksiyonunu ve d değerinin bir çeşitleme parametresini gösterdiği bu ceza yaklaşımı tabu aramada yaygın biçimde kullanılır. Daha fazla çeşitleme etkisi yaratmak için daha büyük d değerleri kullanılır. Örneğin, minimum yayılan ağaç probleminde çözümü daha sık değiştiren düğümler, diğer düğümleri içeren hareketlerin seçimini teşvik amacıyla daha ağır cezalandırılabilir. Negatif cezalar (ödülleri) düşük frekanslı nitelikleri teşvik amacıyla kullanılabilirler. Ceza, hareketlerin niteliklerine uygulandığı gibi hareketlerin sınıflarına da uygulanabilir (Glover, 1990b).

Tabu aramanın ayırt edici başlıca özellikleri (ya da fonksiyonları) şu şekilde özetlenmiştir (Glover, 1995):

Uyarlamalı hafıza:

- Seçicilik (stratejik unutmayı içeriyor)
- Ayırma ve analiz (açık ve niteleyici hafıza vasıtasıyla)
- Zamanlama (olayların hem sıklığını hem de yakınlığını ve kısa dönem ve uzun dönem arasındaki farkı ifade eder)

- Nitelik ve etki (alternatif seçimlerin göreceli cazipliği ve yapıdaki değişimin miktarı anlamında ya da ilişkileri zorlama)
- İçerik (bölgesel, yapısal ve ardışık bağlılık içerme)

Duyarlı arama:

- Stratejik olarak yararlanılan sınırlamalar ve teşvikler (ya da tabu durumları ve amaç düzeyleri)
- İyi alanlara ve iyi çözüm özelliklerine yoğunlaşma (kuvvetlendirme işlemi)
- Umut verici yeni alanları araştırma ve tanımlama (çeşitlendirme işlemi)
- Tek düze olmayan arama şekilleri (stratejik dalgalanmalar)
- Çözümleri tümleştirme ve genişletme (gidiş yolunu yeniden bağlama)

Farklı tabu arama algoritmalarını özel problemleri çözmek için yukarıdaki fonksiyonları birleştirerek formüleştirebiliriz. Tabi ki, asıl geliştirmenin yapıldığı yöntem problem niteliklerine ve kısmi bir uygulamada ihtiyaç duyulan kapsamlılık derecesine dayanır. Yukarıda listelenen fonksiyonlar kümesi genişletilmiş ve /veya değiştirilmiş olabilir, sadece azaltılmış bir fonksiyonlar kümesiyle asıl önerilen yaklaşım, bir çok problemde başarıyla test edilen, kaydetmeye değerdir (tabu listeleri ve amaç kriteri ile kısa dönem hafızalı tabu arama).

3.2.6. Tabu Arama Parametreleri

- Tolerans
- Süreç ve tekrarlama
- Tabu stratejileri
- Uzun dönem stratejiler
- Hassas tabu arama

Tolerans : Kötüleştirmeye ne kadar izin verilir veya iyileştirme ne kadar gereklidir. Eğer sadece geliştirmelere bakılırsa bu tam doğru bir tabu araması değildir. Bu tabu araması ile karşılaştırıldığında önemlidir (Petry ve Wijeysooriya, 2006).

Süreç ve tekrarlama: “Süreç” bir durumun ne kadar tabu halini sürdürdğüdür. “Tekrarlanma” tabu aramanın kaç defa uygun hal listesinde belirlendiğidir (Petry ve Wijeysooriya, 2006).

3.2.7. Tabu Aramanın Temel Stratejileri

3.2.7.1. Riskli (yasaklama) strateji

Tabu listesine hangi hareketlerin girip girmeyeceğine karar veren stratejidir. Stratejinin amacı, döngüyü önlemek ve belirli hareketleri yasaklamak için bir mekanizmanın kurulmasıdır. Döngünün meydana gelme ihtimali mevcut çözümün bir önceki çözümünün bir önceki çözümden uzaklığı ile ilgilidir. Son Tabu Arama, iterasyon boyunca alınan herhangi bir kararın tersini temsil eden hareketlerin seçilmesini önleyerek aramayı geçmişteki çözümlerden uzaklaştırır.

3.2.7.2. Aspirasyon kriteri (Serbest strateji)

Aspirasyon kriteri, araştırma işlemi sırasında karşılaşılan kaliteli çözümlerin kayıp edilmemesi için alınmış önlemler olarak da düşünülebilir. Tabu listesinden neler çıktığının ve ne zaman çıktığının kontrolünü sağlar. Mümkün olduğunda bir hareketin tabu statüsünü geçersiz kılmak için kullanılan bir araçtır. Mevcut durumdan daha iyi bir sonuç verebilecek tabu hareketinin yapılmasına izin verir. Bu kriterle kabul edilen bir hareketin tabu olma durumu kaldırılmaz ve hareket tabu listesinden çıkarılmaz. Sadece hareketin tabu olma durumu aspirasyon kriterini sağladığı için geçici olarak göz ardı edilir. Bir başka ifadeyle, aspirasyon kriteri tabu olan bir hareketin yasak olmasına rağmen yapılabileceği durumları belirtmek için kullanılır. Uygun aspirasyon kriterinin kullanılması, tabu arama performansı üzerinde oldukça etkilidir.

Aspirasyon kriterinin basit bir türü, herhangi bir hareket o ana kadar görülmüş en iyi çözümden daha iyi bir çözüm üretiyorsa yani $f(x_{\text{Tabu}}) < f(x_{\text{eni}})$ ise bu hareketin tabu statüsünün geçersiz kılınmasıdır. Bu şekilde tanımlanan kritere, *Amaca Göre Tümel Aspirasyon* adı verilir ve en yaygın kullanılan kriterdir. Bu yöntemin bir çeşidi de, belli bir bölgedeki en iyi çözümün amaç değerine göre tabu statüsünün iptal edilebildiği *Amaca Göre Yerel Aspirasyondur*.

Tabu Aramada tabu listesi dışında aspirasyon listesi adı verilen ve tek bir değer ihtiva eden bir liste daha vardır. Bu listeye arama boyunca optimizasyon sınırlayıcılarını

sağlayan çözümler kaydedilir. Arama esnasında sınırlayıcıların tasarımını sağlayan bir çözümle karşılaşıldığında bu çözüm aspirasyon listesine kaydedilir. Herhangi bir iterasyonda aspirasyon listesindeki çözümden daha hafif bir çözüm bulunursa bu çözüm tabu olan bir hareket sonucunda elde edilmiş bile olsa kabul edilir ve aspirasyon listesindeki mevcut çözümün yerini alır (Ünler, 2006).

Komşuluk uzayının çok küçük olduğu ya da tabu tutulma süresinin çok yüksek tutulduğu durumlarda, olası bütün hareketlerin tabu aktif eleman içerdiği iterasyonlarla karşılaşılabılır. Bu durumda yukarıda anlatılan kriterden farklı bir aspirasyon kriteri uygulanır. Bu kriter ile, tabu listesi incelenerek en az tabu aktif olan hareketin yapılmasına izin verilir.

3.2.7.3. *Klasik seçme stratejisi*

Bu stratejide, tabu sınırlamaları ve aspirasyon kriteri dikkate alınarak mevcut çözümden, amaç fonksiyonunda en az kötüleşme veya en büyük gelişmeyi üreten hareket seçilir.

3.2.7.4. *İlk ve en iyiyi seçme stratejisi*

Bu stratejide, amaç değeri açısından mevcut çözümden daha iyi bir gelişme kaydeden ilk çözüm seçilir. Gerekli değişiklikler yapılır ve çözümden diğer bir hareketin üretilmesi yapılarak işleme devam edilir. Aday listesindeki (yani komşuluktaki) tüm hareketler bir gelişme kaydetmezse bu strateji en iyi gelişmemiş olan stratejiyi seçer.

3.2.7.5. *Öğrenme stratejisi*

Bu strateji sayesinde Tabu Arama anında bilgi toplanır ve bu bilgi algoritmanın diğer çalışmaları esnasında araştırmayı yönlendirmede kullanılır.

Tabu arama stratejisi, tabu arama işleminin daha genel kısımlarından oluşan bir algoritmadır. TA'nın diğer bazı fonksiyonları; kuvvetlendirme, çeşitlendirme, stratejik dalgalanma, seçilmiş durumlar ve rota birleştirme (path relinking)'dir (Monticelli vd., 2008).

3.2.7.6. *Yoğunlaşma (kuvvetlendirme) stratejisi*

Yoğunlaşma stratejileri geçmişte bulunan iyi çözüm özelliklerini ve hareket kombinasyonlarını teşvik etmek için, değiştirilmiş seçim kurallarını kullanır. Aynı zamanda aramayı daha metodik ve dikkatli yapabilmek için çekici bölgelere bir geri dönüş başlatır. Anlık komşularının gözden geçirilmesi için elit çözümlerin kaydedilmesi

gerektiğinden, açık hafıza, yoğunlaşma stratejilerinin uygulanmasında önemli bir rol oynar (Reeves, 1993).

Yoğunlaşma stratejileri arama sürecinde iyi olarak bulunmuş çözüm özelliklerini ve birleşim hareketlerini destekleyen seçim kurallarını düzenlenmeye dayanır. Aramayı uygun yerlere yönlendirir. İyi çözümler mevcut komşularının araştırılması için kaydedilmelidirler, belirtilmiş hafıza kuvvetlendirme stratejilerinin gelişimi ile yakından ilgilidir (Glover ve Laguna, 1997).

Burada komşuluk kelimesi genel komşuluk aramanın kapsamından daha kapsamlıdır. İlâveten, standart hareket mekanizması ile ifade edilen, iyi çözüme yakın ya da komşu çözümleri göz önünde bulundurur, kuvvetlendirme stratejileri iyi çözümlerin bileşenlerini alarak ya da iyi çözüme götüren gelişmiş değişimleri kullanarak komşular üretir (Glover ve Laguna, 1997).

3.2.7.7. Çeşitleme stratejisi

Çeşitleme stratejileri, ziyaret edilmemiş bölgeleri gözden geçirmek ve daha önce karşılaşılan çözümlerden birçok yönden farklı çözümler yaratmak için arama sürecini teşvik eder. Çeşitleme yaklaşımı da değiştirilmiş değerlendirme stratejilerini kullanabilir. Buna örnek olarak ceza fonksiyonlarının kullanımı verilebilir (Glover, 1995).

3.2.7.8. Stratejik Salınım

Tabu aramada stratejik salınım yaklaşımı, orta ve uzun dönem üzerinden, yoğunlaşma ve çeşitleme arasında anlamlı bir karşılıklı etkileşim sağlar. Stratejik salınım yaklaşımının kullanılmasıyla amaçlanan, aramayı önce seçilmiş bir uygunluk sınırına ve sonra bu sınırdan uzağa doğru yönlendirmektir. Belirtilen sınır, aramanın normalde daha ileri geçemeyeceği bir noktayı gösterir ve kritik düzey olarak da adlandırılır (Sait ve Youssef, 1999).

3.2.7.9. Rota Birleştirme

İki ya da daha fazla elit çözüm kullanılarak yeni bir çözüm elde edilir. Rota birleştirmede kullanılan çözümler referans çözüm olarak adlandırılır. Referans çözümler başlangıç çözümler veya rehber çözümler olarak sınıflandırılabilir. Rota birleştirmeyi

uygulamanın alternatif bir yolu ise tek bir çözüm bulmadan ve oradan yoğunlaştırma veya çeşitlendirme başlatılmasından oluşan yapıcı komşuluk olarak adlandırılır. Başlangıç çözümünün bazı özellikleri seçilir ve her adımda yapıcı bir algoritma rehber çözüme yeni çözümler elde etmek için yeni bir özellik ekler; hem kalitesini arttıran hem de uygulanabilirliğini geliştiren çözümler eklemek gibi. Eğer başlangıç çözümünün çok fazla özelliği mevcutsa, bir yoğunlaştırma işlemi olur. Aksi halde, yeni çözüm çeşitlendirme için temel olarak kullanılacaktır (Monticelli vd., 2008).

3.2.7.10. Elit Durumlar

Elit durumlar, tabu arama algoritmasının arama işlemi boyunca elde ettiği en iyi çözümlerin bir listesidir. Bu elit çözümler diğer çözümlerden ayrı olarak saklanırlar ve yoğunlaştırma ve çeşitlendirme işlemlerinde yeniden başlamak için kullanılabilir. Ancak, bu çözümler en çok rota birleştirmede başlangıç çözümleri veya rehber çözümler olarak kullanılır. İyi kalitenin yanı sıra çözümler arasında çeşitlilik sunduğu için elit çözümler verimli kullanılır. Bu nedenle, elit çözümler listesi kalite ve çeşitlilik dikkate alınarak şekillendirilir (Monticelli vd., 2008).

3.2.7.11. Yeniden Başlama Stratejisi

Tabu arama yönteminde yeniden başlama işlemi, çeşitleme etkisi yaratmak için kullanılan bir stratejidir. Algoritmanın işleyişi periyodik olarak kesilir ve yeni bir çözüm dizisi üretmek için tasarlanmış bir süreçle yeniden başlatılır. Yeniden başlama stratejisinin kullanılma amacı, aramayı çözüm uzayında daha önce ziyaret edilmemiş yeni bölgelere yönlendirmektir (Glover, 1995).

Rastgeleliğe dayanan klasik yeniden başlama stratejileriyle de çeşitleme etkisi yaratmak mümkündür. Buna karşın, daha stratejik yeniden başlatma işlemleri kullanılması, oluşturulacak tabu arama algoritmasını daha etkin kılacaktır. Çeşitlemenin genellikle bir şekilde rastgeleleştirmeye aynı olduğu konusunda gizli bir varsayım vardır. Birçok arama işleminde, çeşitleme etkisi yaratmak için rastgele bir eleman tanımlanması yaygın bir işlemdir, hatta tavlama benzetimi ve genetik algoritmalar temelde böyle bir işleme dayanır. Rastgeleleştirme ve çeşitleme kavramları soyut bir açıdan birbirlerine eşit olarak da görülebilir. Buna karşın, tabu aramada çeşitlilik, belli bir kümenin (örneğin yeni başlangıç çözümlerinin ya da çözüm uzayında belli bir bölgedeki çözümlerin oluşturduğu bir küme) elemanları arasındaki farklılıklar anlamına gelir ve

böyle farklılıkların tespit edilmesi ancak etkili bir arama stratejisiyle mümkündür. Bu nedenle, diğer arama yöntemlerindeki rastgeleleştirmenin popüler kullanımı tabu aramada etkisizdir. Tabu aramada genellikle farklılığın sağlanabilmesi için stratejik bir seçim en iyidir ve rastgele bir seçim en kötüdür. Rastgeleleştirme terimi arama yöntemleri üzerinde çalışan araştırmacılar arasında popüler bir kavramdır ve arama yöntemlerinin çoğunda kullanılmaktadır. Einstein'ın "Tanrı'nın zar atmadığına eminim" sözü popülerliğini yitirmiştir. Araştırmacıların çoğu kör amaçsızlığın yarattığı mucizevî sonuçlara hayranlık duymaktadır. Tabu arama yönteminin işleyişiyle, bahsedilen bu anlayış birbiriyle ters düşer. Rastgeleliği kullanan olasılıksal tabu arama çeşidinde bile, rastgele etkilere bağlı olan elemanlar oldukça sınırlandırılır ve iyi belirlenmiş olasılıklar tarafından rastgeleleştirme kullanımı hafifletilir. Bir başka ifadeyle, rastgeleliğin yer aldığı durumlar dikkatli bir biçimde şekillendirilir. Tabu arama yönteminde zar atılmış olsa bile, atılan zarlar ağırlıklandırılmıştır. Örneğin, rastgeleleştirme yeniden başlama stratejisinin bir parçası olarak kullanıldığında, frekans bilgisi, yeni başlangıç çözümünün üretim sürecini etkileyen olasılık dağılımını tahmin etmek için kullanılabilir. Arama tarihi tarafından bu şekilde yönlendirilen rastgelelik artık bir kör mekanizma olmaktan çıkar (Glover ve Laguna, 1997).

3.2.7.12. Kritik Olay Hafızası

Yeniden başlama işleminin amacı aramanın daha önce gidilmemiş bölgelere ulaşmasını sağlamaktır. Yeniden başlama işlemi belli periyotlarla tekrar tekrar uygulanırken, eğer daha önce aramanın yeniden başlatıldığı bir çözümden arama tekrar başlatılırsa, daha önce incelenmiş olan bölge bir daha incelenmiş olacak yani, kullanılan yeniden başlama stratejisi başarısız olacaktır. Bu nedenle, yeniden başlama stratejisinde tabu aramanın yeniden başlatıldığı yeni başlangıç çözümünün, önceki başlangıç çözümlerinden veya arama boyunca incelenmiş tüm diğer çözümlerden farklı olması istenebilir. Tabu aramada bu çeşitlemeyi gerçekleştirmek amacıyla kullanılan yöntemlerden biri Kritik Olay Hafızasının (Critical Event Memory) kullanımındır . Yeniden başlama stratejisinin başarılı olabilmesi için yeniden başlangıç için seçilen çözüm noktaları arasında farklılık olması gerekmektedir. Bu nedenle, arama sırasında belirli kritik olayların oluşması izlenir ve bu olaylara ait bilgi kritik olay hafızasında saklanır. Daha sonra, elde edilen bu bilgi tabu aramanın yeniden başlatılacağı çözümlerin yaratılmasında kullanılır.

Örneğin, yeni bir başlangıç çözümü aranırken önceki başlangıç çözümleri kritik olayların oluşturulmasında rol oynayabilir. Yeniden başlama işlemi çok kez uygulanırsa, tüm önceki başlangıç çözümleri kritik olay olur ve kritik olay hafızasında saklanır. Daha sonra yeni başlangıç çözümleri oluşturulurken bu kritik olaylar dikkate alınarak, önceki başlangıç çözümlerinden farklı yeni başlangıç çözümleri üretilir. Amaç farklı yeni başlangıç noktalarına bağlı olarak farklı çözüm dizileri elde etmek ve böylece, çözüm uzayının daha önce incelenmemiş bölgelerine ulaşmaktır (Glover, 1996).

Kritik olaylara karşılık gelen çözümlere kritik çözümler denilir ve bu çözümler farklı şekillerde tanımlanabilir. Arama sırasında incelenmiş çözümlerin her birini bir kritik olay olarak tanımlamak bu yollardan biridir, fakat çoğu kez daha küçük bir çözümler kümesini izole etmek de yeterlidir. Örneğin, kritik olay hafızası arama sırasında bulunan yerel optimal, yani amaç fonksiyonu değeri kendinden hemen önceki ve sonraki çözümden daha iyi olan ya da daha kötü olmayan, çözümlerden yararlanılarak da oluşturulabilir (Glover ve Laguna, 1997).

3.2.8. Tabu Arama Bellek Yapıları

Tabu arama son yıllarda birçok karşılaştırmalı çalışmanın ve pratik uygulamaların odağı olmuştur, ve sonuç olarak zor optimizasyon problemlerini çözmek için tercih edilen stratejiler hakkındaki verimli buluşlar ortaya çıktı. Bazen bu buluşların anlamı ve yapısı yeterince açık olmayabilir. Bu belirsizliğin nedeni tabu aramanın literatürde iki farklı şekilde sunulmasıdır. Bunlardan biri tabu aramanın daha basit, diğeri daha ileri düzey olan iki farklı şeklidir. Basit olan yöntem, tabu arama modelinin kısıtlanmış bir parçasını içermekte ve bileşenlerinin sınırlı alt kümelerinin performanslarını test etmek için kullanılan başlangıç analizlerinde kullanılmaktadır (genellikle sadece kısa dönemli hafıza ile ilgilidir). Daha ileri düzey olan yöntem, daha geniş bir yapı şekillendirir. Bu yapı, yoğunlaşma ve çeşitleme stratejileri ile ilgili daha uzun dönemli hafızayı içerir. İkinci yaklaşım, stratejik hafıza bileşenlerinin bir derlemine kullanma üzerine odaklanmasına bağlı olarak, uyumsal hafıza programlaması adını da alır (Glover, 1996).

Tabu Aramanın bellek yapısı 4 ana ilkeye bağlı olarak iş görmektedir (Glover ve Laguna, 2002):

- Çözümün yakın geçmişte incelenmiş olması
- Çözümün incelenme sıklığı
- Çözümün kalitesi
- Çözümün etkisi

Bu basit dizilim şaşırtıcı oranda alternatifleri gizler – bu dört temel ilke kombinasyonlarda göz önünde bulundurulduğunda anlaşılır ve bölgelerin değişimi ve sürenin yayılımı hakkında farklı özellik sınıfları için ayrılmıştır. Etki kavramı, örneğin, tipik olarak bölgesellik, esneklik ve yapıda değişimden söz eder. Ek olarak, bölgesel yapılar tek boyutla sınırlandırılmamış bu ilkelerle ilişki kurmak için kullanılır ama sıralı mantık ve olaya dayalı mantık arasındaki ayrımları davet eder.

Belli başlı bazı Tabu Arama bellek yapıları bulunmaktadır. Bunların dışında çözümü aranan probleme göre bellek yapıları oluşturulabilir. En çok bilinen bellek yapıları:

- Yakın Geçmişe Dayalı Bellek Yapısı
- Sıklığa Dayalı Bellek Yapısı
- Karma Bellek Yapısı

3.2.8.1. Yakın Geçmişe Dayalı Bellek Yapısı

En temel tabu arama bellek yapısıdır. Esas olarak görevi yakın geçmişte yapılan hareketlerin tekrarlanmasını bir süre için tabu olarak işaretlemektir. Yakın Geçmişe Dayalı Bellek Yapısı için iki çok bilinen uygulama mevcuttur :

- Yalnız ters hareketlerin tabu olması
- Gerçekleşen hareketin kendisinin tabu olması

Araştırmanın seyri sırasında yakın geçmişte elde edilen çözümler ile ilgili bilgilerin tutulduğu hafızaya yakınlık tabanlı hafıza adı verilmektedir. Yakın geçmişte ziyaret edilen çözümlerin değişen nitelikleri tabu-aktif olarak ve bu tabu aktif elemanları veya bu elemanların kısmi kombinasyonunu içeren çözümler .tabu. olarak belirlenir. Bu işlem ile yakın geçmişte ziyaret edilmiş çözümlerin $N^*(x)$ içerisine alınmaları, dolayısı ile tekrar ziyaret edilmeleri engellenmiş olur.

3.2.8.2. Sıklığa Dayalı Bellek Yapısı

Sıklığa Dayalı Bellek Yapısı, Yakın Geçmişe Dayalı Bellek Yapısını tamamlayıcı bir özelliğe sahiptir. Genellikle Yakın Geçmişe Dayalı Bellek Yapısı ile birlikte ikinci bir bellek olarak kullanılır.

Sıklığa Dayalı Bellek Yapısı adından da anlaşıldığı gibi yapılan hareketlerin sıklığını bellekte tutar ancak bir hareketin kaç kez yapıldığı bilgisi yerine, çözümün kalitesine ve hareketin etkisine bağlı bilgilerin bellekte tutulmasını daha faydalı yaklaşımlar olarak benimser. Sıklığa Dayalı Bellek Yapısı, Yakın Geçmişe Dayalı Bellek Yapısından farklı olarak hareketin tabu olarak işaretlenmesinden çok, umut verici hareketlerin daha fazla yapılmasını sağlamak için kullanılır (Glover ve Laguna, 2002).

Sıklığa dayalı hafıza tabu aramanın önemli bir özelliğidir. Bu, algoritma tarafından yapılan en sık hareketler esnasında değişen çözüm özelliklerini aklında tutan bir çeşit kısa dönem hafızadır. Tabu-aktif olarak, son zamanlarda ziyaret edilmiş çözümlerin seçilmiş özellikleri etiketlenir. Sıklık ölçümü genellikle bir elemanın değişmeye uğradığı iterasyon sayısı ya da bir elemanın çözümün bir parçası olarak yer aldığı iterasyon sayısı olarak belirlenir. Bir elemanın değişime uğradığı veya bir elemanın çözümün bir parçası olduğu iterasyon sayısını tutan sıklık ölçümleri birbirleriyle alakalı bilgi içermelerine karşı farklı manalar taşımaktadır. Bir elemanın çok sayıda çözümde var olması, kaliteli çözümlere ulaşılmışsa o elemanın değerli bir eleman olduğunu gösterir. Bir elemanın değişime uğradığı iterasyon sayısı çok fazlaysa ve bir süredir araştırmada ilerleme gözlenememişse bu elemanın, etkisi düşük bir eleman olduğu anlaşılır. Böyle bir elemanın hareket etmesi engellenerek, araştırmanın seyrinin değişmesi sağlanabilir. Yüksek frekans değeri olan elemanlar bir tabu aktivasyon kuralını tetikleyebilirler ve tabu listesine alınarak tekrarlanmaları engellenebilir. Literatürde belirtilen pratik uygulamaların çoğu direk olarak bu temel TA algoritmasını uygulamaktadır. Bu, en temel TA algoritması çeşididir, yasaklanan özellikler listesine ve amaç kriterine dayanır. TA listesinin temel amacı döngüyü engellemek ve komşuluğu azaltmaktır (tabu olarak etiketlenen durumları çıkararak). TA listesi kullanmanın asıl dezavantajı, yasaklanan özelliklerin incelenmemiş etkili komşulukların bir parçası olabileceğidir. Bu problemin üstesinden gelmek için bir tabu durumu ile

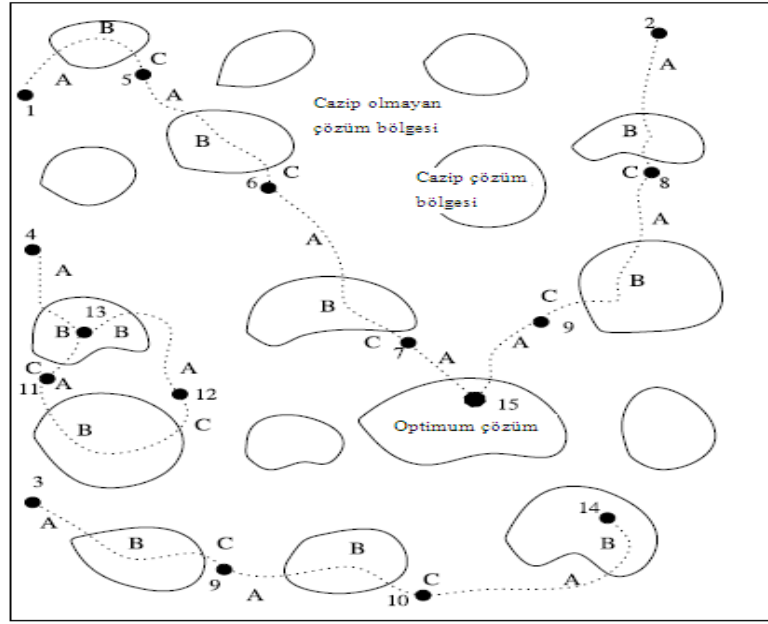
ilgili maliyet o ana kadarki en düşük maliyetten daha düşükse tabu durumu ortadan kalkar ve deęişime izin verilir (Monticelli vd., 2008).

Yüksek kalitede çözümler içeren bir bölgede, elde edilen büyük bir kalıcılık frekansı değeri çok çekici bir nitelięi işaret edebilir ya da eęer ilgilenilen bölge düşük kalitede çözümler içeriyorsa bunun tersini işaret edebilir. Dięer taraftan, eęer ilgilenilen bölge hem yüksek, hem de düşük kalitede çözümler içeriyorsa, büyük ya da küçük bir kalıcılık frekansı aramanın kısıtlanmasına neden olan yerleşik (ya da hariç) bir nitelięi işaret edebilir. Buna göre, çeşitleme sağlamak ve aramanın bu bölgede sıkışıp kalmasını engellemek amacıyla, sözü edilen yerleşik nitelięin kullanılmamasına (ya da kullanılmasına) karar verilmesinde, elde edilen bu kalıcılık frekansı kullanılır. Yüksek bir geçicilik frekansı ise amaç fonksiyonunun daha iyi değerler alabilmesi için çözümlere dahil olan ya da çözümlerden çıkan bir nitelięin varlıęının belirlenmesinde yardımcı olabilir (Glover, 1995).

Sıklık ölçütünün dört unsuru vardır (Glover ve Laguna, 2002):

- Her bir hareketin toplam tekrar sayısı
- Toplam hareket sayısı
- En yüksek hareket tekrar sayısı
- Ortalama hareket sayısı

Şekil 3.8 yukarda anlatılan tip kısa dönem TA algoritmasının çalışmasını gösterir. Şekilde dört farklı yol gösterilmiştir: 1-5-6-7-15 yolu ile 2-8-9-15 yolu optimal çözüme ulaşır; 3-9-10-14 yolu yerel optimuma yakalanır; 4-11-12-13 yolu ise döngü oluşturur. Tabu sınırlamaları zorlansa bile, eęer tabu aktif olduęu sırada k hareket sayısı oransal olarak küçükse döngü meydana gelebilir. k'nın aşırı büyük değerleri ise ziyaret edilen mevcut cazip durumlar gecikmeye yol açıp aramayı etkin olmayan bir hale getirebilirler. Şekil 3.8'de gösterilmeyen alternatif yol ortadaki nokta olarak optimal çözüme sahip olabilir. Bu durumda çözüm işlemi optimal çözümden geçer ve optimal olmayan bir çözümde durana kadar devam eder. Bu ciddi bir problemde gösterilmez, çünkü optimal çözüm hafızada mevcut en iyi çözüm olarak saklanır (Monticelli vd., 2008).



Şekil 3.8 : Tabu Aramanın kullanıldığı bir arama örneği (Monticelli vd., 2008)

3.2.8.3. Karma Bellek Yapısı

Bir tabu arama algoritmasında hem Sıklığa Dayalı Bellek Yapısı hem de Yakın Geçmişe Dayalı Bellek Yapısı bulunmasına rağmen birbiriyle bağımlı iki bellek listesinin olması durumunda oluşan bellek yapısına Karma Bellek Yapısı denebilir. Bu işlem Sıklığa Dayalı Bellek Yapısının pekiştirici ve çeşitlendirici işlemleri yerine getirebilmesi için önerilen bir tekniktir (Glover ve Laguna, 2002).

3.2.9. Programlama ve Sunumun Etkisi

Monticelli vd. (2008)'ne göre diğer tümleşik yaklaşımlarda olduğu gibi sunum ve programlama tabu aramanın anahtar konularıdır. Amaç fonksiyonunun belirlenmesi kadar önemli olan uygun ve uygun olmayan bölgelerin tanımlanması direk olarak çözülen problem modelinde kullanılan sunumun ve programlamanın tipine bağlıdır. Sunum ve programlamada kullanılan metodun etkilediği diğer kritik konu verilen durumun komşuluğunun belirlenmesidir. Tabu aramanın temel fonksiyonları ile ilgili üç bakış açısı şunlardır:

- çözüm uzayında yapılan değişimler;
- arama uzayındaki her çözüm nasıl belirtilir
- verilen bir durumun komşuluğu nasıl tanımlanır.

Bu görüşler ileride belirtilecek grafiksel açıklamanın yardımıyla en iyi şekilde anlaşılır. Arama uzayı şekil 3.8'de optimal çözüm içeren, cazip olan ve olmayan bölgelerin gösterildiği gibi canlandırılabilir. Tabu arama bir noktadan başlar, Şekil 3.8'de 1 noktasında olduğu gibi, ve optimal çözüme doğru hareket eder.

3.2.10. Tabu Arama Algoritmasında Hafıza Kullanımı

Tabu arama yönteminin temel dayanak noktası, aşağıda verilen üç hafıza bileşeninin etkili biçimde kullanımınıdır (Glover, 1990a):

- 1) Kısa dönem hafıza bileşeni: Tabu arama yönteminin çekirdeğini oluşturur.
- 2) Orta dönem hafıza bileşeni: Aramayı bölgesel olarak yoğunlaştırmak için kullanılan hafıza bileşenidir.
- 3) Uzun dönem hafıza bileşeni: Aramada tümel olarak çeşitleme etkisi yaratmak için kullanılan hafıza bileşenidir.

3.2.10.1. Kısa Dönem Hafıza

Tabu arama, çok genel bir sezgisel arama metodudur. Bu nedenle bu metodun uygulanışı ile ilgili kesin prensipler yoktur. Tabu arama metodundan daha yüksek verim alabilmek için algoritmaya, çözülmesi istenen problemin yapısına göre eklemeler yapılması çoğu kez zorunlu olmaktadır. Bu nedenle, tabu arama ile bir probleme çözüm aranırken, kullanılacak hafıza, probleme uygun şekilde tasarlanmalıdır (Reeves, 1993).

Kısa dönem hafıza bileşeninin ana fikri, belirli arama yönlerini tabu olarak sınıflandırmak ve böylece yerel optimal tuzaklarından aramayı uzak tutmaktır. En çok kullanılan kısa dönemli hafıza, yakın geçmişte değişen çözüm niteliklerini ya da çözümlerin kendisini saklar ve yenilik temelli hafıza olarak adlandırılır. Aspirasyon kriteri sağlandığında ise çözümlerin ya da niteliklerin tabu durumu iptal edilir. Sadece kısa dönem hafıza kullanan tabu arama yöntemi, araştırma hakkında uzun dönem hafıza kullanan tabu arama kadar geniş bilgi sahibi olmasa da bir çok probleme kaliteli çözüm bulabilmesi ve kolay uygulamaya geçirilmesi açısından oldukça tercih edilen bir yöntemdir. Aşağıda tabu aramanın temel basamakları verilmektedir (Reeves, 1993):

1. Başlangıç çözümü oluştur,
2. Komşu çözümler üreterek bu çözümlerden uygun bir tanesini seç,

3. Seçilen çözüm ile mevcut çözümün yerini değiştir ve tabu listelerini yenile,
4. Durdurma kriteri sağlanana kadar adım 2 ve adım 3'ü tekrarla.

Kısa dönem hafızanın kullanımında, karşılaşılan durumların bir ayırt edici H tarihi, arama sürecini yönetmek için saklanır. $N(x)$ komşu kümesi yerine H tarihinin bir fonksiyonu olan ve $N(H,x)$ ile gösterilen bir değiştirilmiş komşuluk kümesi kullanılır. x çözümünden bir hareketle hangi çözüme ulaşılabileceğini tarih belirler çünkü x ' yeni çözümü, $N(H,x)$ komşuluğundan seçilir.

3.2.10.2. Orta dönem hafıza

Tabu arama algoritmasında, orta dönem hafıza bileşeninin temel rolü, umut vaat eden bölgelerde aramanın yoğunlaşmasını sağlamaktır. Yoğunlaşma stratejisi tabu arama yönteminin daha iyi sonuçlara ulaşmasını sağlayan önemli bileşenlerinden biridir. Yoğunlaşma stratejisi kullanımıyla arama daha agresifleşir. Yoğunlaşma stratejileri seçim kurallarını değiştirerek, geçmişte iyi olarak bulunan çözüm özelliklerini ve hareket kombinasyonlarını ödüllendirirler. Ayrıca bu stratejiler çekici bölgeleri daha iyi inceleyebilmek için bir geri dönüş başlatırlar. Tabu aramada genel olarak yoğunlaşma stratejisinin işleyişi aşağıda verildiği gibi tanımlanmıştır (Glover ve Laguna, 1997):

Adım-1 Kısa dönem hafızalı tabu aramayı çalıştır.

Adım-2 Elit seçim stratejisini uygula.

Adım-3

(A) En iyi deneme çözümlerinden birini seç.

(B) Seçilen çözümü başlangıç çözümü olarak kısa dönemli tabu aramayı yeniden çalıştır.

(C) Yeni bulunan uygun ve iyi çözümleri elit listesine ekle.

(D) İterasyon sayısı önceden belirlenmiş bir sınırı aşarsa ya da elit listesi boşsa DUR, aksi halde Adım_3 (A) dan tekrar devam et.

Orta dönem hafıza bileşeninde, aramanın belirli bir periyodu esnasında yaratılan en iyi deneme çözümlerinden seçilmiş n tanesi kaydedilir. Örneğin, seçilecek bu çözümler, ardışık çözümlerden n tane en iyisi ya da arama esnasında ulaşılan n tane yerel optimal çözüm olabilir. İyi olarak nitelendirilen bu seçilmiş çözümler ya da çözümlerin belli niteliklerini içeren yeni çözümler üzerinde yoğunlaşarak, daha iyi bir amaç değerine ulaşılmaya çalışılır (Sait ve Youssef, 1999).

3.2.10.3. Uzun Dönem Hafıza

Bir çok uygulamada kısa dönem hafıza bileşenleri, kaliteli çözümler üretmede yeterli olmaktadır. Uzun dönem hafıza stratejilerinin de katılmasıyla tabu aramanın gücü belirgin şekilde artmaktadır. Başka bir deyişle, daha iyi çözümlerin bulunma ihtimali, kısa dönem hafıza ile beraber uzun dönem hafızanın da kullanılması ile artmaktadır. Sıklık tabanlı hafıza, uzun dönem hafıza stratejilerinde kullanılan bir hafızadır. Bununla araştırmanın sadece belirli bir dönemine ait bilgiler değil, araştırmanın bütünü ile ilgili bilgiler tutulabilir. Araştırmanın izlediği seyir incelenerek, başarılı çözümlerin etrafında yer alabilecek daha iyi çözümler ortaya çıkarılabilir veya başarılı olmadığı anlaşılan hareketler cezalandırılarak tekrar bu hareketlerin yapılması engellenir. Uzun dönem hafıza stratejilerinden yoğunlaştırma stratejileri ile metod, mevcut çözümü önceden iyi olarak belirlenmiş çözüme benzeyecek şekilde hareket seçimi yapmaya zorlanır. Birbirine çok benzer çözümlerin oluşmasını engellemek amacıyla yoğunlaştırma içinde bazı değişiklik ölçümleri uygulanabilir. Örneğin elit çözümler kayıt edilirken, sadece birbirinden belirgin derecede farklı olan çözümler kayıt edilebilir. Ve her bir elit çözüm incelenmek üzere seçildiğinde önceki çözüme ait kısa dönem hafıza boşaltılabilir. Bir başka yoğunlaştırma metodu da elde edilmiş iyi çözümlerin parçalarının birleştirilerek çözüm elde etmeye çalışmaktır.

Araştırmanın küçük bir alanda dolaşmasını engellemek için değişiklik stratejileri kullanılır. Genetik algoritmalarda çeşitliliğin sağlanması için rastgelelik prensibiyle çalışan çaprazlama ve mutasyon operatorleri kullanılır. Tabu arama metodunda kısa dönem hafıza fonksiyonları ile de değişiklik sağlanabilir ancak bu işlem uzun dönem hafızanın da eklenmesi ile güçlendirilir. Tabu aramada uzun dönem hafıza bileşenin amacı aramanın farklı bölgelere yönlendirilmesiyle çeşitleme etkisinin yaratılabilmesidir. Tabu aramanın yerel optimal tuzağından kurtulmasını sağlayan bileşenlerinden biri çeşitlendirme stratejisidir. Bu nedenle, çeşitlendirme etkisinin yeterli ölçüde yaratılabilmesi algoritmanın optimale yakın çözümlere ulaşabilmesinde büyük önem taşır. Bir hareket seçilirken, bu hareketin nadir yapılmış ya da daha önceden hiç yapılmamış bir hareket olması için seçim kuralları zorlanarak önceki çözümlerden daha farklı özellikte çözümler elde edilmesi sağlanabilir. Burada bağlı kalınan ilkeler, orta dönem hafıza fonksiyonu tarafından kullanılan ilkelerle ters düşer. Orta dönem hafıza bileşeninde önceden bulunmuş iyi çözümleri içeren bölgelere

aramanın daha yođun olarak odaklanması sađlanır. Uzun dnem hafıza bileşeninde ise arama, Őimdiye kadar incelenen blgelerden farklı yeni blgelere ynlendirilir (Reeves, 1993).

Aramanın czm uzayında henz ulařamadıđı yeni blgeleri keřfetmek amacıyla cřitlendirme stratejileri kullanılır. Sezgisel arama yntemleri genellikle, bir kombinatoryal optimizasyon problemi tarafından tanımlanan czm uzayını keřifte etkilerini arttırabilmek iin, cřitlendirme stratejilerine gvenirler. Bu stratejilerden bazıları dngden kaınmak, yani arama srecinde durmadan aynı hareket dizilerinin yapılmasını ya da daha genel olarak devamlı aynı czm kmelerinin yeniden ziyaret edilmesini engellemek amacı ile geliřtirilmiřlerdir. Diđerleri ise aramayı daha fazla sađlamlařtırmak iin ileri srlmüş stratejilerdir. Cřitleme stratejisi kullanılmadıđında, arama czm uzayının kk bir alanında yerelleřir ve tmel optimal bir czm bulma olasılıđı ok zayıflar (Sait ve Youssef, 1999).

Arama sırasında nadiren kullanılan ya da hi kullanılmayan czmleri ya da bu czmlerin niteliklerini kullanabilmek iin seim kurallarının deđiřtirilmesi, tabu arama cřitlendirme stratejilerinin temelini oluřturur. Tabu aramada cřitlendirme stratejileri bu az kullanılan ya da kullanılmayan niteliklerin alt kmelerini aday czmlere ekleyen yntemleri tekrarlı olarak kullanabilir ya da czm srecini kısmi veya tam olarak yeniden bařlatabilirler. Tabu aramada cřitlendirme stratejilerinin zellikle etkin oldukları nokta, daha iyi czmlere ulařmak iin czm uzayındaki tmseklere ya da engellerin ařılmasını sađlamalarıdır (Glover, 1995).

Yenilik temelli tabu kriterinin tersine, sıklık temelli iřleyen bir uzun dnem hafıza, uzun dnemli bir tabu listesini ya da herhangi bir diđer uygun veri yapısını kullanabilir (Sait ve Youssef, 1999).

Tercih edilen hareketlerin seimi ařamasında yararlanılan deđerlendirme esasları, sıklık temelli hafıza kullanımıyla geniřletilmiř olur. Yenilik temelli hafızadan elde edilen bilgiyi tamamlayıcı trde bir bilgi elde edilmiř olur. Yenilik temelli yaklařımda olduđu gibi, czm kalitesi ve hareket etkisi hesaba katılır ve buna gre czmn bulunma

sıklığı çoğunlukla ağırlıklandırılır ya da alt sınıflara ayrıştırılır (Glover ve Laguna, 1997).

3.2.11. Tabu Aramanın Diğer Yöntemlerle Karşılaştırılması

- Geleneksel azalan metodu, gelişmemiş hareketlere izin vermez, TA ise verir.
- Tavlama Benzetimi ve Genetik Algoritma örnekleme kullanan yarı rasgele işlemlere dayanır, TA deterministik yapıdadır.
- TB ve GA belirgin hafızaya sahip değildir, TA ise sahiptir.
- Dal-Sınır Algoritması sabit hafıza taslağına sahiptir (sabit dallanma stratejisi), TA esnek hafızaya sahiptir
- TA kötü bir stratejik seçimin iyi bir rasgele seçimden daha verimli bilgi sağladığı görüşünü savunur (Glover ve Laguna, 2002).

3.2.12. Tabu Arama Uygulamaları

Tabu arama algoritmasının sık kullanıldığı diğer alanlara ise şu örnekler verilebilir (Glover ve Laguna, 2002):

- Planlama
- Tasarlama
- Yer belirleme ve atama
- Mantık ve yapay zeka
- Teknoloji
- Telekomünikasyon
- Üretim, stok ve yatırım
- Rotalama
- Grafik iyileştirme

3.3. KARESEL ATAMA PROBLEMİ

Üretim ve hizmet sektöründeki verimlilik üzerinde uzun dönemli etkisi ile stratejik bir konu olan Tesisiçi Yerleşim Problemi (TYP), yaygın olarak üzerinde çalışılan bir problemdir. TYP, Kuadratik Atama Problemi (KAP), karesel ayarlama seti problemi (quadratic set covering problem), lineer tam sayılı programlama ve karışık teorik problem içeren bir çok formülasyonla modellenmiştir. TYP için çözüm yöntemleri;

kesin yaklaşımlar, sezgiseller, meta-sezgiseller ve diğerleri olarak sınıflandırılabilir (Chiang vd., 2011).

TA'nın TYP uygulamalarına örnek çalışmalar vardır: W. C. Chiang ve Kouvelis (1996), yakın geçmişe dayalı ve uzun dönem hafıza yapılarını, dinamik tabu listesi boyutu stratejisini, yoğunlaştırma ve çeşitlendirme stratejilerini içeren bir TA uygulaması KAP problemi çözümünde kullanmışlardır. W.C. Chiang ve C. Chiang (1999), TA, olasılıklı TA, Benzetimli Tavlama ve melez TA içeren KAP formülasyonunu ile TYP çözümüne uygulamışlardır. Chitratamawat ve Noble (1999) bir TA yaklaşımı ile yerleşim düzenini, yükleme boşaltma alanlarını ve malzeme taşıma ekipmanlarını aynı anda göz önünde bulunduran entegre bir problem çözümü geliştirmişlerdir. McKendall ve Hakobyan (2010) ise TA kullanan bir dinamik bir yerleşim problemini tartışmışlardır. Sürekli bir düzlemsel alanda rastgele veya manüel olarak üretilen başlangıç yerleşimi ile birlikte program tesisteki değişim ve hareketlerin uygulaması için bir TA algoritması kullanır ve yaklaşımlar yavaş yavaş en iyi çözüme ya da olası optimum çözüme gider.

Tesisi düzenlenme sorununun çıkışı, her bölümün ilgili konumunu belirten bir blok yerleşimidir. Daha sonra, bölümlerin kesin yerlerini, koridor yapılarını, giriş-çıkış noktalarını ve her bölüm için bölüm içindeki düzeni ayrıntılı olarak belirten yerleşim düzeni elde etmek için daha fazla çalışma gerçekleştirilebilir. Ayrıntılı yerleşim problemi; akış hattı yerleşimi problemleri, makine yerleşimi problemleri ve makinelerin eşit alan kapladığının ve aynı boyutlarda olduğunun varsayıldığı hücreli imalat tasarımı problemlerini içerir (Meller, 1996).

Literatürde, yerleşimin etkinliği genellikle malzeme taşıma maliyetleri olarak ölçülür. Bu maliyetler aşağıdaki parametrelerin bir veya daha fazlası ile ilgilidir; bölümler arası akışlar, f_{ij} (i bölümünden j bölümüne akış); birim maliyet değerleri, c_{ij} (i bölümünden j bölümüne bir birim yükü bir birim mesafeye taşımanın maliyeti) ve bölümler arası yakınlık dereceleri, r_{ij} (i ve j bölümleri arasındaki yakınlığın sayısal değerleri). Bu parametreler iki ortak malzeme taşıma maliyeti fonksiyonlarında kullanılır (Meller, 1996).

Bunlardan ilki bölümlerin yan yana olup olmadığına dayanır (Meller, 1996);

$$\max \sum_i \sum_j (r_{ij}) x_{ij}$$

Eğer i ve j bölümleri bitişik ise x_{ij} 1'e eşittir, yoksa 0'dır. Böyle bir amaç, iki bölüm bitişik olduğunda malzeme taşıma maliyetinin önemli ölçüde azalacağını belirten malzeme taşıma prensibine dayanır.

Diğer malzeme taşıma maliyet fonksiyonu ise bölümlere ait uzaklıklara dayanır (Meller, 1996);

$$\min \sum_i \sum_j (f_{ij} * c_{ij}) d_{ij}$$

d_{ij} i bölümünden j bölümüne olan uzaklıktır. Bu amaç, malzeme taşıma maliyetlerinin birim yükün taşındığı mesafeye göre artacağını belirten malzeme taşıma prensibine dayanır. Bu uzaklıklar çeşitli yöntemlerle çözülür; en çok kullanılan iki yöntem (Meller, 1996):

- Giriş-çıkış noktaları arasındaki uzaklık: bu uzaklık iki bölüm arasındaki giriş-çıkış noktalarını belirtir ve bazı durumlarda iki bölüm arasında taşıma yaparken koridor boyunca ölçüm yapılır. Bu ölçümün büyük bir dezavantajı ise, detaylı ölçüm yapıp geliştirilmedikçe giriş-çıkış noktalarının (ya da koridorların) yerlerinin bilinmiyor olması durumunda kullanılamayıp, yaygın olarak kullanılan ağırlık merkezi yaklaşımına yönlendirmesidir.
- Merkezden merkeze (ağırlık merkezi) uzaklık: bölümlerin giriş-çıkış noktalarının bilinmediği durumlarda bölüm merkezi bölümün bölümün giriş-çıkış noktasını gösterir. Ağırlık merkezine dayanan bir algoritma bölümlerin merkezlerini mümkün olduğunca yakın olarak hizalamaya çalışır, bu bölümü çok uzun ve dar yapabilir ve L şeklinde bir bölümün ağırlık merkezi bölümün dışında kalabilir.

Yukarıda bahsedilen her bir uzaklık ölçümü için iki nokta arasındaki mesafeyi ölçmede kullanılan iki ölçüm vardır. Doğrusal mesafe en çok kullanılan uzaklıktır, çünkü dik (ortogonal) eksenlere paralel yollar boyunca hareket etmeye dayanır. İkinci uzaklık ise iki noktayı birleştiren doğrusal bir yol boyunca ölçülen uzaklıklar için uygun olan öklid uzaklığıdır (Meller, 1996).

KAP formülasyonu her bölümü bir konuma ve her konuma en az bir bölüm atar. Bir bölümü belirli bir konuma yerleştirmenin maliyeti etkileşimli bölümlerin konumuna bağlıdır. KAP'nin son zamanlarda ortaya atılan yeni bir alternatif formülasyonu bölümler arası uzaklıkların bölüm eşleşmelerine atanmasını göz önünde bulundurur (Meller, 1996).

KAP için çözüm teknikleri, başlangıçta Gilmore ve Lawler tarafından tasarlanan dalsınır yaklaşımlarının çeşitliliğine dayanır. KAP, genel olarak çözümü zor olan bir NP-tam problemidir. Bugüne kadar, bu problemin genel durumlar için optimal çözümleri yalnızca 18'den az bölüme sahip problemler için bulunabilir. Kaku, Thompson, Morton, Smith ve MacLoed tarafından yapılan çalışmalarda görülebilir (Meller, 1996).

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 problemlerde bile bazı zorluklarla karşılaşmıştır. Bu sorun, bazı melez uygulamalarla giderilip etkili sonuçlar elde edilmiştir.

Burkard ve Rendl (1984), KAP'in ilk tavlama benzetimi uygulamalarından birini gerçekleştirmişlerdir.

Dorigo vd. (1996) başta olmak üzere, son yıllarda yapılan çalışmalar; yakın iyi çözümlere sahip örnekler başta olmak üzere, karınca kolonisi en iyileşmesinin diğer üst sezgiseller karşısında rekabetçi üstünlüğe sahip olduğunu göstermiştir. Karınca kolonisi en iyileşmesinin KAP'ne uygulanmasını içeren en önemli çalışmaların başında Gambardella vd. (1999)'in olduğu söylenebilir.

Liao, Kusiak ve Heragu ve diğerlerine göre eşit alanlara sahip olmayan tesis yerleşimi düzenlenmesi, bölümleri eşit alanlı küçük karelere bölünmesi bölünmemiş aynı bölümlerin kareleri arasında büyük bir yapay akış sağlaması ve KAP sonuçlarının çözümü ile değiştirilen bir KAP olarak modellenabilir. Ancak bölümlerdeki artış nedeniyle birkaç eşit olmayan alana sahip bölümlerin olduğu küçük çaptaki problemlerde bile çözüm mümkün değildir. Üstelik, Bozer ve Meller böyle bir

yaklaşımın sınırlı bölümler ekleyeceğinden dolayı verimli olmayacağını göstermişlerdir. KAP' ne dayalı gelişen formülasyonlar yerleşimle ilgili son çalışmalarda iyice gözden geçirilmiştir (Meller, 1996).

Temel Fabrika Yerleşim Problemi olarak da bilinen Sınırsız Kapasiteli Tesis Yerleşim Problemleri (SKTYP), TYP ailesinin bir üyesidir. SKTYP, NP-Zor yapıya sahip olduğundan dolayı, problem boyutu ne kadar artarsa optimal sonucu bulmakta o kadar zordur ve ayrıca güvenilir sonuçlara ulaşmak da bir hayli zaman gerektirir (Yiğit ve Türkbey, 2003).

TYP, karesel optimizasyon problemleri içerisinde önemli bir yere sahip olup, çözüm zamanının, problem boyutuna bağlı olarak üstel artış gösterdiği çözülmesi zor NP-Hard problemlerdendir. Bu tür problemler, problem boyutunun polinom fonksiyonu ile sınırlı olan hesaplama zamanı içinde çözülemezler. Bu nedenle bu tür problemlerin çözümünde, optimumu bulmayı garanti etmeyen ancak, çözüm zamanı polinom sınırlar içinde kalan sezgisel metotlardan yararlanır. Sezgisellerin çoğu probleme özgü özelliğe sahiptirler ve bir problem için kullanılan bir sezgisel, başka bir problem için kullanılamamaktadır. Bununla beraber, problemlere uygulanmaları açısından esnek ve daha genel özelliğe sahip olan tekniklere (Tavlama Benzetimi, Tabu Arama, Yapay Sinir Ağları, Genetik Algoritmalar) ilgi son yıllarda giderek artmıştır. Bu teknikler çok sayıda problemlere uygulanmışlar ve oldukça güçlü bulunmuşlardır (Yiğit ve Türkbey, 2003).

TYP, bir tesisin verimli bir şekilde çalışması için tesisteki fiziki bölümlerin düzenlenmesi ile ilgilenir. TYP, disiplinler arası önemi nedeniyle pek çok araştırmaya konu olmuştur. TYP için iyi bir çözüm işlemlerin etkinliğinin bütününe katkı yapar. Kötü bir yerleşim ise, süreç içi stokların birikmesine, malzeme taşıma sistemlerinin aşırı yüklenmesine, hazırlık zamanlarının artmasına ve malzeme kuyruklarının uzamasına sebep olur. Bu nedenle, TYP'nin çözülmesi, bütün işletmeler için yapılması gereken stratejik bir çalışmadır (Şahin ve Türkbey, 2010).

Genel olarak, TYP'ni çözmek için iki temel yaklaşım kullanılmaktadır. Birincisi bölümler arasındaki yakınlık ilişki skorlarının toplamını en büyülemeye çalışan nitel

yaklaşımıdır. Muther tarafından geliştirilen Sistematik Düzenlenme Planlaması (Systematic Layout Planning) prosedürü, bu konudaki en önemli çalışmalardan birisidir. İkincisi ise, bölümler arasındaki malzeme taşıma maliyetlerinin toplamını en küçükleme yaklaşımıdır (Şahin ve Türkbey, 2010).

TYP genellikle Kareli Atama Problemi (KAP) olarak modellenir. KAP, bölümler arasındaki malzeme taşıma maliyetlerinin toplamını en küçükleyecek şekilde N adet bölümün N adet alana atamasıyla ilgilenir. KAP'nin en iyi çözümünü bulmak için birçok yöntem önerilmiştir. Ancak KAP'nin NP-zor yapısı nedeniyle, kabul edilebilir zaman dilimi içinde şimdiye kadar en fazla 30 bölümden oluşan problemler için en iyi çözüm bulunabilmiştir. Bu nedenle, problemin çözümü için daha çok sezgisel algoritmalar (genetik algoritma, tavlama benzetimi, tabu arama, karınca kolonisi sistemi v.b.) kullanılmıştır. Sezgisel yöntemler en iyi sonucu garanti etmemelerine rağmen, kısa zaman diliminde en iyiye yakın sonuçlar bulabilmektedirler (Şahin ve Türkbey, 2010).

3.3.1. KAP Tabanlı Tesis Yerleşimi Problemleriyle İlgili Çalışmalar

KAP, tesis yerleşim problemlerinde en iyi çözümü elde etmek için geliştirilmiş geleneksel yaklaşımlardan biridir. Tesis yerleşimi probleminin bu formülasyonunun, orta büyüklüklerdeki problem için 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 (Utkan, 2006).

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 (Utkan, 2006).

Lacksonen ve Enscore (1993), dinamik tesis yerleşim problemlerini, farklı bölüm boyutları kullanılabilir şekilde genişletilmiş KAP ile modellemiştir. Çalışmada; CRAFT, kesme düzlemleri, dal sınır algoritması, dinamik programlama ve kesme ağaçları algoritmaları bu model kullanılarak dinamik tesis yerleşimi problemleri için karşılaştırılmıştır (Utkan, 2006).

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 (Utkan, 2006).

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 karşılaştırılmıştır. Çalışma sonucunda elde edilen veriler, tesis tasarımı ve yerleşim problemleri için sezgiselleri kullanmanın rekabetçi avantaj sağladığını ortaya çıkartmıştır.

Tahir vd. (2007) hem 1-0 hem de kuadratik programlama örneği olan numune sınıflandırması için özellik seçimi ve ağırlıklandırılması problemini TA yaklaşımıyla K-en yakın komşu kuralına göre çözmüştür (Cura, 2008).

3.3.2. Açıklayıcı Basit Bir Problem ve Çözümü

Kuadratik problemlerin çözümünde p 'nin n 'li permütasyonunun formu şeklinde gösterilebilir. Genellikle bir problem için kullanılan işlemin çeşidi 2 işlem elemanı olan i ve j 'nin yerini değiştirmektir. Bununla beraber çözüm için uygulanabilir bütün hareket adımlarının verimli bir biçimde bulunması mümkündür. KAP çözümü için n elemanın atanmasında sıralama p ile gösterilmiştir. i ve j bölümler olarak tanımlanmış ve bu bölümlerin yer değişimi şeklindeki atama hareketleri bu problem tipi için sıklıkla kullanılmıştır (Dreo vd., 2006).

Kısa vadede aramayı yönlendirmek için kullanılan bir teknik de t iterasyonları esnasında devam edilen hareketlerin terslerinin alınmasının engellenmesidir. Eğer harekete

permütasyon p ile başlanırsa, hareketin tersi eş zamanlı olarak i elemanı p_i kısmında, j elemanı p_j kısmında yer alacak şekilde bir hareket tanımlanır, fakat bu yöntem dönüşümü korumak için en etkili yöntemlerden biridir ve hareketin tersine dönmekten sakınan t parametresi en az duyarlı olanlardan biri olarak görünür (Dreo vd., 2006).

Yapılan bir hareketin tersinin yapılması t iterasyon süresince kısa dönem hafıza sayesinde yasaklanmıştır. Eğer p sıralamasında (i, j) hareketi uygulanmışsa bu hareketin tersi i bölümünün P_i sitesine, j bölümünün P_j sitesine aynı anda yerleştirilmesi şeklinde tanımlanmıştır (Dreo vd., 2006).

t 'nin düzenlemiş değerleri güçlü bir arama üretmez, çünkü dönüşümler t 'nin büyük değerleri için görünebilir. Bu problemi çözmek için bahsedildiği gibi t rassal olarak tek tek ve $(0,9.n)$ ve $(1,1.n+1)$ arasında olacak şekilde çizilir. Ayrıca denemeler gösteriyor ki bir tabu araması süresi problemin genişliğine eşittir ya da küçük örnekler için daha az verimli olmasına neden olacak şekilde biraz fazla olur. Statik bir durum için ideal olacak değerden hafif miktarda fazla değer seçmektense t 'nin değerini arama işlemi süresince hareketli bir yöntemle seçmek daha etkin bir yol olabilir (Dreo vd., 2006).

Bu tabu mekanizmasını pratikte uygulamak için bir T matrisi kullanılır, t_{ir} i bölümünün r sitesinden son kez yer değiştirip p_i sitesine gittiği tabu süresi t 'ye eklenen iterasyon numarasını verecektir. Böylece, hem t_{ipi} ve t_{jpi} girişleri mevcut iterasyondan daha büyük bir değere sahipse (i, j) hareketi yasaklanacaktır (Dreo vd., 2006).

Literatürde NUG5 olarak bilinen 5×5 boyutunda küçük bir KAP probleminin tabu arama ile çözümünü gösteren bir örnektir. Bu örnekte yer alan 5 alan $(1, 2, 3, 4, 5)$ ve 5 bölüm $(1,2,3,4,5)$ bulunmaktadır ve bu bölümler bu alanlara atanmaktadır. D alanlar arasındaki uzaklığı, F ise bölümler arasındaki malzeme akışını göstermektedir (Dreo vd., 2006).

1	3	
2	4	5

Şekil 3.9 :KAP çözümü için örnek bir yerleşim

$$\mathcal{F} = \begin{pmatrix} 0 & 5 & 2 & 4 & 1 \\ 5 & 0 & 3 & 0 & 2 \\ 2 & 3 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 5 \\ 1 & 2 & 0 & 5 & 0 \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} 0 & 1 & 1 & 2 & 3 \\ 1 & 0 & 2 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 \\ 2 & 1 & 1 & 0 & 1 \\ 3 & 2 & 2 & 1 & 0 \end{pmatrix}$$

İterasyon 0: $P(i, j)$, i ve j bölümlerinin yer değiştirmesini gösterir. $\Delta P(i, j)$ ise i ve j bölümlerinin yer değiştirmesi durumunda oluşan taşıma maliyetindeki değişim miktarını gösterir. T başlangıç matrisi 0'dan başlar, yani $T=0$ 'dır.

İterasyon 1: Rassal olarak belirlenen başlangıç çözümü için 1, 2, 3, 4, 5 alanlarına sırasıyla 2, 4, 1, 5, 3 bölümleri atanır, bu durum $p=(2, 4, 1, 5, 3)$ şeklinde gösterilir. Bu çözümün maliyeti 72 birim olarak hesaplanır.

Hareket	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
Maliyet	2	-12	-12	2	0	-10	-12	4	8	6

(1, 3), (1, 4) ve (2, 5) sitelerindeki bölümlerin yer değişimi maliyetteki maksimum azalmayı sağlamıştır. Bunlardan herhangi biri seçilebilir, sıralamaya göre seçilecek olursa (1, 3) hareketi, yani 1 ve 3 sitelerindeki 1 ve 2 bölümlerinin yer değişimi gerçekleşir. $t=9$ olursa 10. iterasyona kadar 1 numaralı siteye 2 numaralı bölümün atanması, 3 numaralı siteye ise 1 numaralı bölümün atanması yasaklanmış olur.

$$\mathcal{T} = \begin{pmatrix} 0 & 0 & 10 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

İterasyon 2: 1. iterasyonda seçilen hareketin maliyet değişimi 12 birimdir, yeni sıralama $p=(1, 4, 2, 5, 3)$ olur ve bu durumda toplam maliyet 60 birim olmuştur. Elde edilebilecek her yer değişimi değeri bu durum için yendiden hesaplanır.

Hareket	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
Maliyet	14	12	-8	10	0	10	8	12	12	6
Tabu		evet								

Bu iterasyon için, bir önceki hareketin tersinin yasaklandığını görebiliriz. (1, 4) hareketi bize 8 birim maliyet azalması ile minimum maliyeti verir. Eğer hareketin tersinin rassal olarak seçilen tabu süresi $t=6$ olursa, T matrisi şu şekilde olur:

$$T = \begin{pmatrix} 8 & 0 & 10 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

İterasyon 3: $p=(5, 4, 2, 1, 3)$ çözümünün maliyeti 52 birim olmuş ve yerel optimuma ulaşmıştır. Aslında, 3. iterasyonun başında maliyeti azaltacak bir değişim olmamıştır.

Hareket	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
Maliyet	10	24	8	10	0	22	20	8	8	14
Tabu			evet							

Bu iterasyonda seçilen (2, 3) hareketi 0 birim maliyet değişimine sahiptir. Burada 2. iterasyonda yasaklanmış olan (1, 3) hareketinin 5 numaralı bölümün daha önce hiç 3 numaralı siteye atanmadığı için bu değişimin yeniden çözüme girdiğini görüyoruz. Eğer rassal olarak seçilen tabu süresi $t=8$ olursa, T matrisi şu şekilde olur:

$$T = \begin{pmatrix} 8 & 0 & 10 & 0 & 0 \\ 10 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

İterasyon 4: elde edilen $p=(5, 2, 4, 1, 3)$ çözümünün maliyeti 52 birimdir.

Hareket	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
Maliyet	24	10	8	10	0	8	8	22	20	14
Tabu			evet		evet					

Ancak, minimum maliyeti sağlayan (2, 3) hareketini seçmek bizi önceki çözümlere döndürebileceğinden dolayı yasaklı olduğu için mümkün değildir. Benzer bir durum, daha önce ziyaret edilmemiş bir çözüme götürecek bir değişim olmasına rağmen, 1 numaralı siteye yerleştirilen 5 numaralı bölüm ve 4 numaralı siteye yerleştirilen 1 numaralı bölümü gösteren (1, 4) hareketi için de söz konusudur. Bu nedenle, daha önce seçilmemiş olan ve maliyeti 8 birim arttıran (2, 4) hareketini seçeriz. Tabu süresi $t = 5$ olarak seçilirse:

$$T = \begin{pmatrix} 8 & 0 & 10 & 9 & 0 \\ 10 & 9 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

İterasyon 5: Bu iterasyonun başında çözüm $p=(5, 1, 4, 2, 3)$, çözüm maliyeti ise 60 birimdir. Hareketlerin maliyetleri hesaplanırsa:

Hareket	(1,2)	(1,3)	(1,4)	(1,5)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)
Maliyet	12	-10	12	10	0	-8	4	14	20	10
Tabu			evet		evet	evet				

Burada görülüyor ki; daha önceki çözümlerde çözümün kalitesini azaltan bir hareket olan (1, 3) hareketi şimdi optimal çözüm $p=(4, 1, 5, 2, 3)$ 'e ulaşıp çözüm maliyetini 50 birime düşürerek yararlı hale gelebiliyor.

4. BULGULAR

4.1. PROBLEMLE İLGİLİ VERİLER

Tabu arama algoritması, literatürde yer alan C. E. Nugent, T. E. Volland ve J. Ruml (1968) tarafından yapılan çalışmalarla elde edilmiş, NUG test problemleri olarak bilinen tesis yerleşimi düzenlenmesi ile ilgili test problemleri üzerinde denenmiştir. NUG test problemleri, KAP için literatürde en çok kullanılan problemlerden biridir, her departman için uzaklık ve iş akış matrisleri içerir. $n = \{14, 16, 17, 18, 21, 22, 24, 25\}$ olan problemler J. Clausen vd. (1997) tarafından daha fazla deparmana sahip büyük örneklerden, 27 ve 28 departman içeren örnekler ise Anstreicher vd. tarafından (2000) 30 departman içeren örnekten satırlar ve sütunlar silinerek elde edilmiştir. Kullanılan bu problemlere internet üzerinden erişilmiştir (<http://www.opt.math.tu-graz.ac.at/qaplib/inst.html>). Tablo 4.1’de NUG problemlerinin optimum sonuçları verilmiştir.

KAP çözümü için kullanılan tabu arama algoritması genel olarak aşağıdaki adımlardan oluşur.

- 1- Program değişkenleri tanımla (Başlangıç çözümü, komşuluk yapısı, tabu süresi büyüklüğü, iterasyon sayısı, amaç fonksiyonu vb.)
- 2- Başlangıç çözümü rassal olarak bulunur ve maliyet hesapla
- 3- Tabu listesini başlat
- 4- Mevcut çözümün komşuları bulunur ve maliyetleri hesapla
- 5- Çözümler değerlendirilir ve en iyi maliyetli çözüm seç
- 6- Komşuluktaki en iyi çözüm mevcut çözümden daha iyi değilse 4. adıma dön. Daha iyi ise çözümün tabu olup olmadığına bak, tabu değil ise mevcut çözüm yap ve 4. adıma dön, tabu ise aspirasyon kriteri sağlanıyormu bak, sağlanmıyorsa bir sonraki en iyi komşuluğun değerini al, sağlanıyor ise mevcut çözüm yap ve 4. adıma dön.
- 7- Tabu listesini güncelle.

8- 4. Adımdan 7 adıma kadar olan adımları iterasyon sayısı tamamlanana kadar tekrarla iterasyon tamamlanınca aramayı sonlandır.

Tabu arama algoritmasının tez konumdaki problem çözümü için yazılan program kodları:

Değişken Tanımları

Dim depSayisi As Integer

' Departman sayısı değişkenleri

Dim a() As Integer

' Departman sayısı dizileri

Dim EnIyiMaliyet As Long

' Optimum maliyet değişkeni

' Program değişkenleri

Dim UzunHafıza() As Integer, UzunDeger As Integer

Dim TabuElemanSayisi As Integer

Dim TabuListesi(10000, 4) As Integer

Dim tasima() As Integer

Dim uzaklik() As Integer

Dim maliyet() As Integer

Dim optimum As Integer

Rassal Çözüm Algoritması

ReDim a(depSayisi) As Integer

ReDim tasima(depSayisi, depSayisi) As Integer

ReDim maliyet(depSayisi, depSayisi) As Integer

ReDim uzaklik(depSayisi, depSayisi) As Integer

For i = 1 To depSayisi

For j = 1 To depSayisi

tasima(i, j) = Sheet3.Cells(2 + i, 1 + j)

uzaklik(i, j) = Sheet4.Cells(2 + i, 1 + j)

*maliyet(i, j) = tasima(i, j) * uzaklik(i, j)*

Next j

Next i

Optimum Maliyet Hesaplama Algoritması

Private Function MaliyetHesapla(a() As Integer) As Long

Dim i As Integer, j As Integer

Dim tasima2 As Integer

Dim uzaklik2 As Integer

Dim m As Long, t1 As Integer, t2 As Integer, m2 As Long

```

For i = 1 To depSayisi
  For j = 1 To depSayisi
    If j <> i Then
      m2 = m2 + tasima(a(i), a(j)) * uzaklik(i, j) ' maliyet(a(i), a(j))
    End If
  Next j
Next i
MaliyetHesapla = m2
End Function

' En iyi tabu olmayan çözümün seçilmesi
Private Sub TabuArama()
Dim iterNum As Integer, k As Integer, i As Integer
Do
  iterNum = iterNum + 1
  Call KomsulukBak(iterNum)
  For i = 1 To TabuElemanSayisi
    If TabuListesi(i, 3) = 1 And TabuListesi(i, 4) < iterNum - 100 Then
      TabuListesi(i, 3) = 0
    End If
  Next i
  If UzunDeger <= optimum Then Exit
Do Loop Until iterNum = 5000
Cells(4, 2) = UzunDeger
For k = 1 To depSayisi
  Cells(5, 1 + k) = UzunHafiza(k)
Next
End Sub

'tüm komsuluklari hesapla
For i = 1 To depSayisi
  For j = i + 1 To depSayisi
    temp = a(i)
    a(i) = a(j)
    a(j) = temp
    s = s + 1
    m(s, 1) = MaliyetHesapla(a)
    m(s, 2) = i
    m(s, 3) = j
    temp = a(i)
  
```

```

    a(i) = a(j)
    a(j) = temp
  Next j
Next i
'bulunanlari sirala
Do
deg = False
For i = 1 To s
  For j = i + 1 To s
    If m(i, 1) > m(j, 1) Then
      tem(1) = m(j, 1)
      tem(2) = m(j, 2)
      tem(3) = m(j, 3)
      m(j, 1) = m(i, 1)
      m(j, 2) = m(i, 2)
      m(j, 3) = m(i, 3)
      m(i, 1) = tem(1)
      m(i, 2) = tem(2)
      m(i, 3) = tem(3)
      deg = True
    End If
  Next j
Next i
Loop Until deg = False
'küçükten büyüye bakarak bulunan ilk uygun degeri kabul et
'j sırayla küçükten büyüye bak
bulundu = False
For i = 1 To TabuElemanSayisi
  If TabuListesi(i, 3) = 1 And m(j, 2) = TabuListesi(i, 1) And m(j, 3) = TabuListesi(i, 2) Then
    bulundu = True
    bulundu_no = i
  End If
Next i
'güncel çözüm yap
temp = a(m(j, 2))
a(m(j, 2)) = a(m(j, 3))
a(m(j, 3)) = temp
'optimum çözüm yap
For k = 1 To depSayisi

```

$$UzunHafıza(k) = a(k)$$

Next k

$$UzunDeger = m(j, 1)$$

Algoritma Visual Basic dilinde kodlanmış, Excel programında makro olarak çalıştırılmıştır ve 2 GB ram, Pentium (R) Dual-Core CPU 2.10 Ghz işlemcili bir bilgisayarda test edilmiştir. $n = \{12, 14, 15, 16, 17, 18, 20, 21, 22, 24, 25, 27, 28, 30\}$ olmak üzere farklı departman sayıları (n) içeren 5, 25 ve 100 olan farklı tabu sürelerine ve 500, 1000 ve 5000 olan farklı iterasyon sayılarına sahip 15 adet problem üzerinde çözülmüştür.

Tablo 4.1: NUG Problemleri için Literatürde Bulunan Optimum Sonuçlar

Departman Sayısı	Optimum	KAP
12	578	(12,7,9,3,4,8,11,1,5,6,10,2)
14	1014	(9,8,13,2,1,11,7,14,3,4,12,5,6,10)
15	1150	(1,2,13,8,9,4,3,14,7,11,10,15,6,5,12)
16-a	1610	(9,14,2,15,16,3,10,12,8,11,6,5,7,1,4,13)
16-b	1240	(16,12,13,8,4,2,9,11,15,10,7,3,14,6,1,5)
17	1732	(16,15,2,14,9,11,8,12,10,3,4,1,7,6,13,17,5)
18	1930	(10,3,14,2,18,6,7,12,15,4,5,1,11,8,17,13,9,16)
20	2570	(18,14,10,3,9,4,2,12,11,16,19,15,20,8,13,17,5,7,1,6)
21	2438	(4,21,3,9,13,2,5,14,18,11,16,10,6,15,20,19,8,7,1,12,17)
22	3596	(2,21,9,10,7,3,1,19,8,20,17,5,13,6,12,16,11,22,18,14,15)
24	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)
25	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)
27	5234	(23,18,3,1,27,17,5,12,7,15,4,26,8,19,20,2,24,21,14,10,9,13,22,25,6,16,11)
28	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)
30	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)

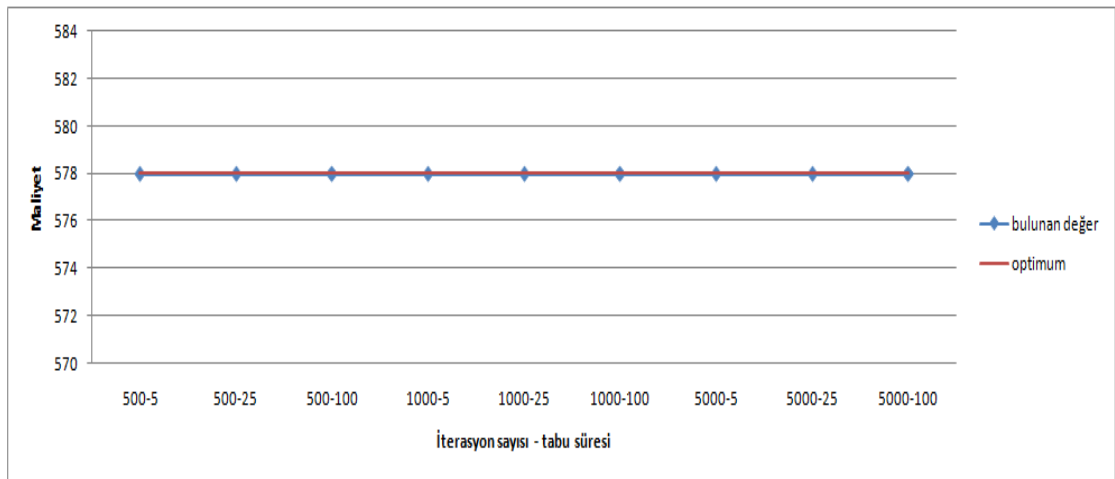
4.1.1. NUG12 Problemi için Elde Edilen Sonuçlar

NUG12 problemi 12 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.2'de, optimum sonuçlarla karşılaştırılması ise Şekil 4.1'de görülmektedir.

Literatürde Nugent vd. tarafından bulunan en iyi maliyet 578 birimdir. Kullanılan algoritma ile optimuma farklı yerleşim sıralamaları ile ulaşılabilmekte ve yaklaşık 50-200 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.2: 12 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi													Maliyet
500	5	3	9	7	12	1	11	8	4	2	10	6	5	578
500	25	5	6	10	2	4	8	11	1	12	7	9	3	578
500	100	5	6	10	2	4	8	11	1	12	7	9	3	578
1000	5	2	10	6	5	1	11	8	4	3	9	7	12	578
1000	25	12	7	9	3	4	8	11	1	5	6	10	2	578
1000	100	2	10	6	5	1	11	8	4	3	9	7	12	578
5000	5	12	7	9	3	4	8	11	1	5	6	10	2	578
5000	25	3	9	7	12	1	11	8	4	2	10	6	5	578
5000	100	3	9	7	12	1	11	8	4	2	10	6	5	578



Şekil 4.1 :12 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

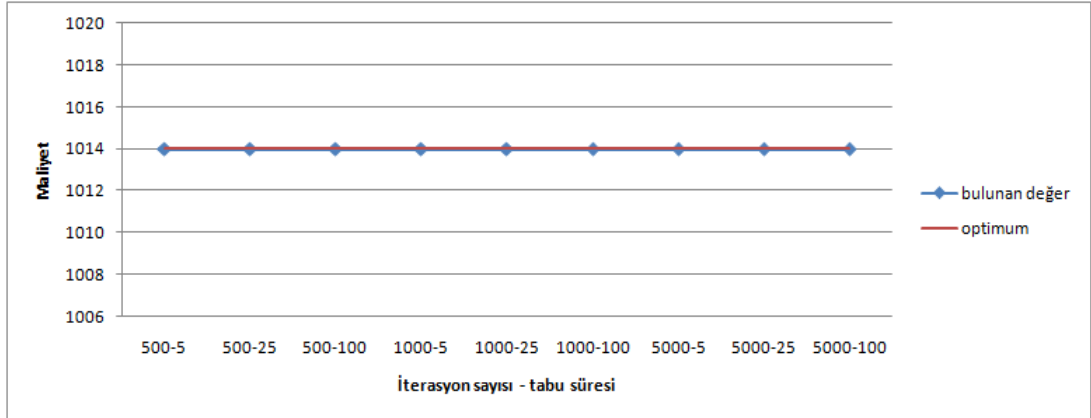
4.1.2. NUG14 Problemi için Elde Edilen Sonuçlar

NUG14 problemi 14 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.3'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.2'de görülmektedir.

Literatürde en iyi maliyet 1014 birimdir. Bu optimal sonuç Clausen and Perregaard (1997) tarafından elde edilmiştir. Kullanılan algoritma ile optimuma tek bir yerleşim sıralaması ile ulaşılabilmekte ve yaklaşık 80-240 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.3: 14 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi															Maliyet
500	5	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
500	25	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
500	100	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
1000	5	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
1000	25	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
1000	100	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
5000	5	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
5000	25	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014
5000	100	9	8	13	2	1	11	7	14	3	4	12	5	6	10	1014



Şekil 4.2: 14 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

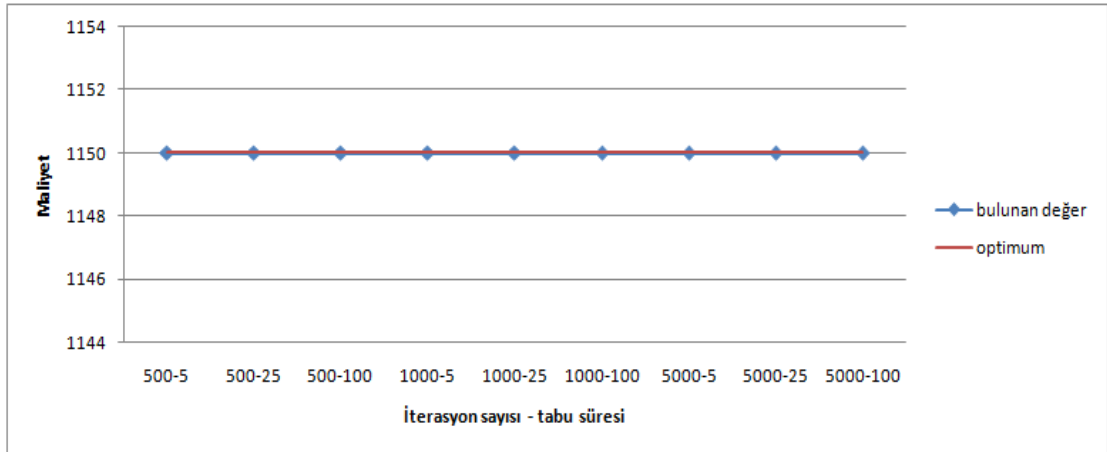
4.1.3. NUG15 Problemi için Elde Edilen Sonuçlar

NUG15 problemi 15 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.4'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.3'de görülmektedir.

Literatürde Nugent vd. tarafından bulunan en iyi maliyet 1150 birimdir. Kullanılan algoritma ile optimuma farklı yerleşim sıralamaları ile ulaşılabilmekte ve yaklaşık 120-300 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.4: 15 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi															Maliyet	
500	5	9	8	13	2	1	11	7	14	3	4	12	5	6	15	10	1150
500	25	9	8	13	2	1	11	7	14	3	4	12	5	6	15	10	1150
500	100	12	5	6	15	10	11	7	14	3	4	9	8	13	2	1	1150
1000	5	1	2	13	8	9	4	3	14	7	11	10	15	6	5	12	1150
1000	25	10	15	6	5	12	4	3	14	7	11	1	2	13	8	9	1150
1000	100	10	15	6	5	12	4	3	14	7	11	1	2	13	8	9	1150
5000	5	12	5	6	15	10	11	7	14	3	4	9	8	13	2	1	1150
5000	25	1	2	13	8	9	4	3	14	7	11	10	15	6	5	12	1150
5000	100	9	8	13	2	1	11	7	14	3	4	12	5	6	15	10	1150



Şekil 4.3: 15 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

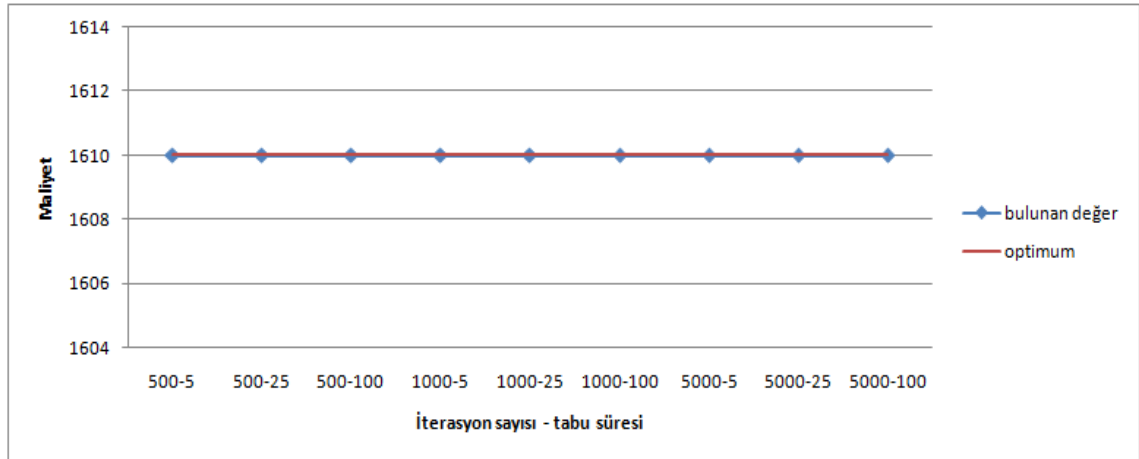
4.1.4. NUG16a Problemi için Elde Edilen Sonuçlar

NUG16a problemi 16 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.5'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.4'te görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 1610 birimdir. Kullanılan algoritma ile optimuma tek bir yerleşim sıralaması ile ulaşılabilmekte ve yaklaşık 210-480 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.5: 16-a Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi															Maliyet		
500	5	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
500	25	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
500	100	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
1000	5	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
1000	25	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
1000	100	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
5000	5	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
5000	25	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610
5000	100	9	14	2	15	16	3	10	12	8	11	6	5	7	1	4	13	1610



Şekil 4.4: 16 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması (a)

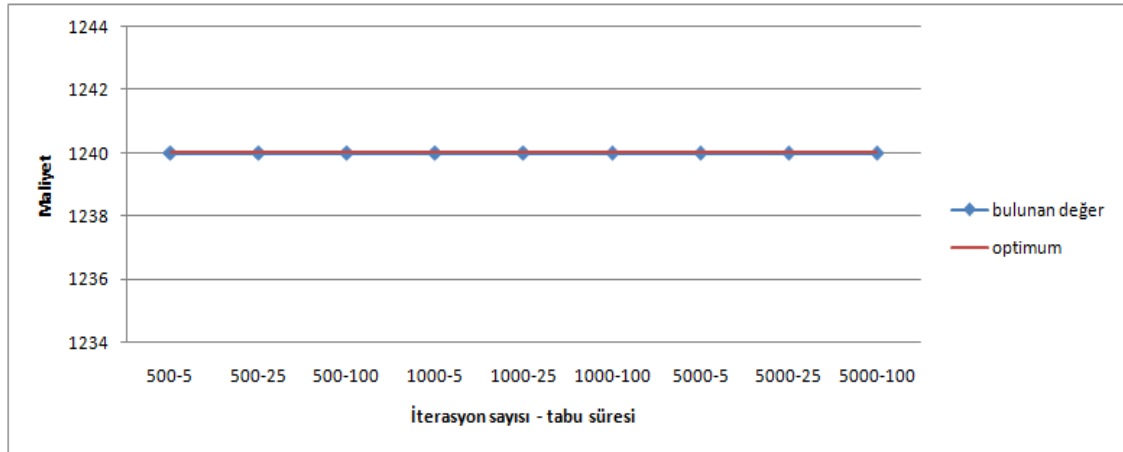
4.1.5. NUG16b Problemi için Elde Edilen Sonuçlar

NUG16b problemi 16 departmana sahip, farklı uzaklık ve iş akış matrislerine sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.6'da, optimum sonuçlarla karşılaştırılması ise Şekil 4.5'de görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 1240 birimdir. Kullanılan algoritma ile optimuma farklı bir yerleşim sıralamaları ile ulaşılabilmekte ve yaklaşık 210-460 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.6: 16-b Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																Maliyet	
500	5	16	4	15	14	12	2	10	6	13	9	7	1	8	11	3	5	1240
500	25	16	12	13	8	4	2	9	11	15	10	7	3	14	6	1	5	1240
500	100	14	15	4	16	6	10	2	12	1	7	9	13	5	3	11	8	1240
1000	5	5	3	11	8	1	7	9	13	6	10	2	12	14	15	4	16	1240
1000	25	16	4	15	14	12	2	10	6	13	9	7	1	8	11	3	5	1240
1000	100	8	13	12	16	11	9	2	4	3	7	10	15	5	1	6	14	1240
5000	5	5	1	6	14	3	7	10	15	11	9	2	4	8	13	12	16	1240
5000	25	5	1	6	14	3	7	10	15	11	9	2	4	8	13	12	16	1240
5000	100	8	13	12	16	11	9	2	4	3	7	10	15	5	1	6	14	1240



Şekil 4.5: 16 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması (b)

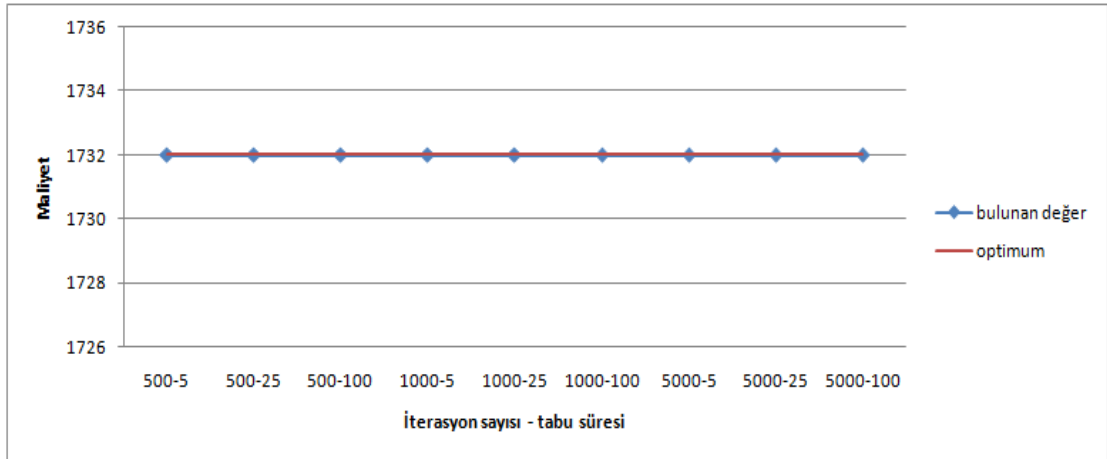
4.1.6. NUG17 Problemi için Elde Edilen Sonuçlar

NUG17 problemi 17 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.7'de, optimum sonuçlarla karşılaştırılması ise Şekil 4.6'da görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 1732 birimdir. Kullanılan algoritma ile optimuma tek bir yerleşim sıralaması ile ulaşılabilen ve yaklaşık 270-660 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.7: 17 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																Maliyet		
500	5	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
500	25	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
500	100	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
1000	5	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
1000	25	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
1000	100	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
5000	5	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
5000	25	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732
5000	100	16	15	2	14	9	11	8	12	10	3	4	1	7	6	13	17	5	1732



Şekil 4.6: 17 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

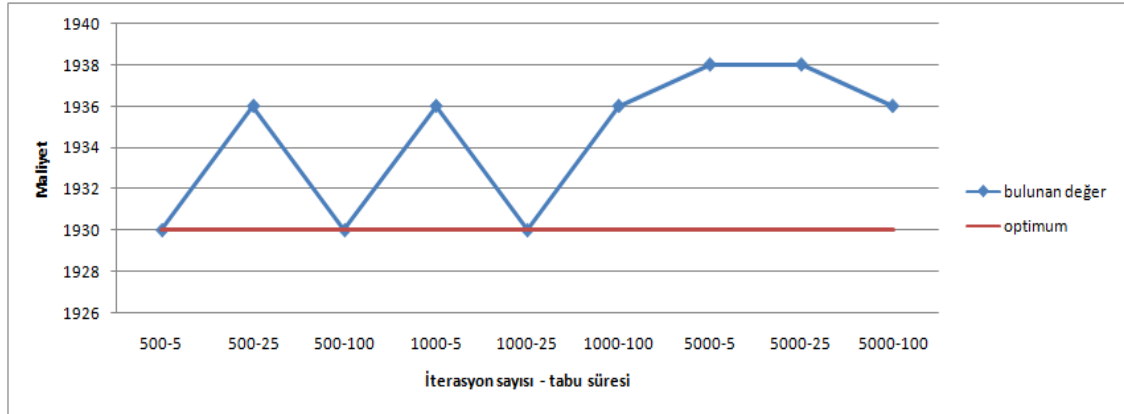
4.1.7. NUG18 Problemi için Elde Edilen Sonuçlar

NUG18 problemi 18 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.8'de, optimum sonuçlarla karşılaştırılması ise Şekil 4.7'de görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 1930 birimdir. Kullanılan algoritma ile bazı parametrelerde optimum değere farklı sıralamalar ile ulaşılabilirken diğer parametrelerde optimuma çok yakın değerler bulunmakta ve bu sonuçlar yaklaşık 330-870 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.8: 18 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																			Maliyet
500	5	10	3	14	2	18	6	7	12	15	4	5	1	11	8	17	13	9	16	1930
500	25	17	1	7	5	6	11	8	12	10	13	4	15	2	3	9	16	18	14	1936
500	100	9	3	10	6	13	14	2	12	7	5	18	15	8	1	17	16	4	11	1930
1000	5	17	1	7	5	6	11	8	12	10	13	4	15	2	3	9	16	18	14	1936
1000	25	9	3	10	6	13	14	2	12	7	5	18	15	8	1	17	16	4	11	1930
1000	100	9	3	10	13	6	14	2	12	7	5	18	15	8	1	17	16	4	11	1936
5000	5	18	2	14	3	9	8	15	12	10	6	4	11	7	5	13	17	16	1	1938
5000	25	18	2	14	3	9	8	15	12	10	6	4	11	7	5	13	17	16	1	1938
5000	100	17	1	7	5	6	11	8	12	10	13	16	15	14	3	9	4	18	2	1936



Şekil 4.7: 18 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

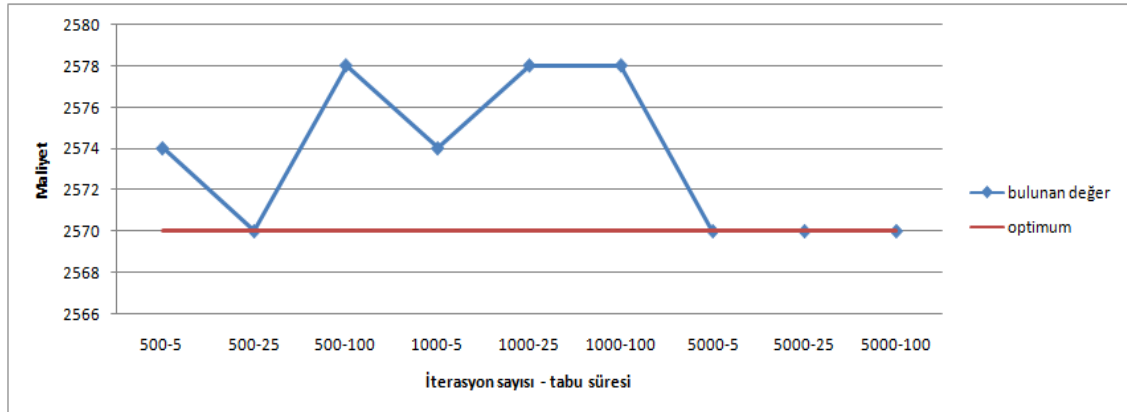
4.1.8. NUG20 Problemi için Elde Edilen Sonuçlar

NUG20 problemi 20 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.9'da, optimum sonuçlarla karşılaştırılması ise Şekil 4.8'de görülmektedir.

Literatürde Nugent vd. tarafından bulunan en iyi maliyet 2570 birimdir. Kullanılan algoritma ile bazı parametrelerde optimum değere farklı sıralamalar ile ulaşılabilirken diğer parametrelerde optimuma çok yakın değerler bulunmakta ve bu sonuçlar yaklaşık 420-1080 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.9: 20 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																			Maliyet		
500	5	18	2	14	3	9	19	15	12	10	6	4	20	7	5	13	17	8	11	1	16	2574
500	25	6	1	7	5	17	13	8	20	15	19	16	11	12	2	4	9	3	10	14	18	2570
500	100	9	3	14	10	18	16	11	12	2	4	13	8	20	15	19	6	1	7	5	17	2578
1000	5	18	2	14	3	9	19	15	12	10	16	4	20	8	11	13	17	7	5	1	6	2574
1000	25	6	1	7	5	17	13	8	20	15	19	16	11	12	2	4	9	3	14	10	18	2578
1000	100	18	10	14	3	9	4	2	12	11	16	19	15	20	8	13	17	5	7	1	6	2578
5000	5	9	3	10	14	18	16	11	12	2	4	13	8	20	15	19	6	1	7	5	17	2570
5000	25	9	3	10	14	18	16	11	12	2	4	13	8	20	15	19	6	1	7	5	17	2570
5000	100	17	5	7	1	6	19	15	20	8	13	4	2	12	11	16	18	14	10	3	9	2570



Şekil 4.8: 20 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

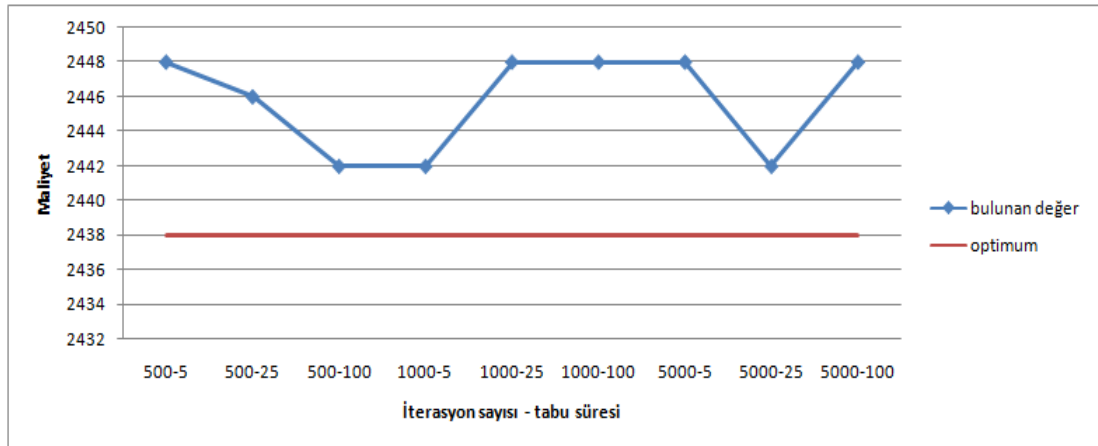
4.1.9. NUG21 Problemi için Elde Edilen Sonuçlar

NUG21 problemi 21 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.10'da, optimum sonuçlarla karşılaştırılması ise Şekil 4.9'da görülmektedir.

Literatürde Bruengger vd. (1996) tarafından bulunan en iyi maliyet 2438 birimdir. Kullanılan algoritma ile optimum sonuç elde edilemese de optimuma çok yakın değerler bulunmakta ve bu sonuçlar yaklaşık 540-1240 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.10: 21 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																					Maliyet	
500	5	5	2	18	21	19	3	20	17	12	1	10	7	8	14	15	6	13	9	16	11	4	2448
500	25	20	3	21	9	10	2	5	14	19	8	7	1	12	17	4	18	11	16	13	6	15	2446
500	100	4	3	21	9	13	2	5	14	18	11	16	10	6	15	20	19	8	7	1	12	17	2442
1000	5	17	12	1	7	8	19	20	15	6	10	11	16	18	14	5	2	13	9	3	21	4	2442
1000	25	15	6	13	9	16	11	4	17	12	1	10	7	8	20	5	2	18	21	3	19	14	2448
1000	100	4	11	16	9	13	6	15	20	8	7	10	1	12	17	14	19	3	21	18	2	5	2448
5000	5	14	19	3	21	18	2	5	20	8	7	10	1	12	17	4	11	16	9	13	6	15	2448
5000	25	4	3	21	9	13	2	5	14	18	11	16	10	6	15	20	19	8	7	1	12	17	2442
5000	100	5	2	18	21	19	3	20	17	12	1	10	7	8	14	15	6	13	9	16	11	4	2448



Şekil 4.9: 21 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

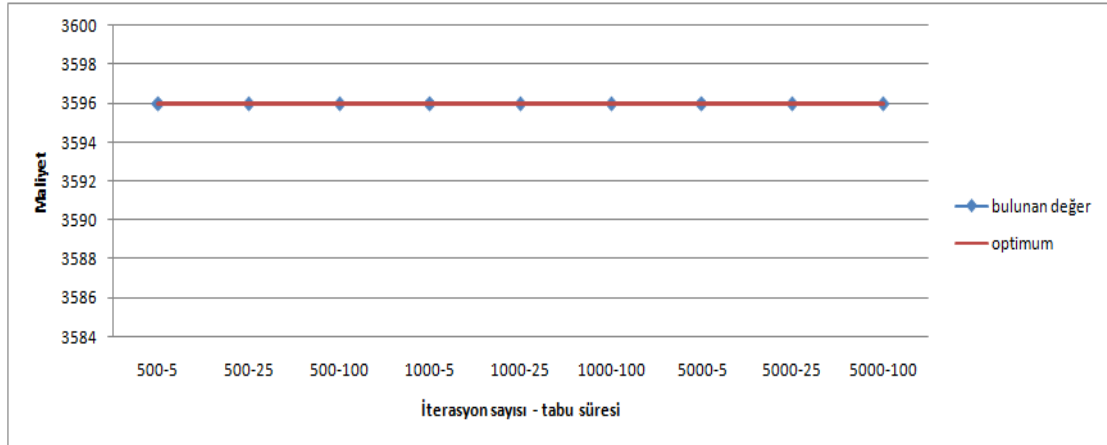
4.1.10. NUG22 Problemi için Elde Edilen Sonuçlar

NUG22 problemi 22 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.11'de, optimum sonuçlarla karşılaştırılması ise Şekil 4.10'da görülmektedir.

Literatürde Bruengger vd. tarafından bulunan en iyi maliyet 3596 birimdir. Kullanılan algoritma ile optimuma farklı bir yerleşim sıralamaları ile ulaşılabilmekte ve yaklaşık 630-1480 saniyelik bir sürede optimum sonuç elde edilebilmektedir.

Tablo 4.11: 22 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																					Maliyet		
500	5	2	21	9	10	7	3	1	19	8	20	17	5	13	6	12	16	11	22	18	4	14	15	3596
500	25	2	21	9	10	7	3	1	19	8	20	17	5	13	6	12	16	11	22	18	4	14	15	3596
500	100	15	14	4	18	22	11	16	12	6	13	5	17	20	8	19	1	3	7	10	9	21	2	3596
1000	5	17	20	8	19	1	3	7	10	9	21	2	15	14	4	18	22	11	16	12	6	13	5	3596
1000	25	5	13	6	12	16	11	22	18	4	14	15	2	21	9	10	7	3	1	19	8	20	17	3596
1000	100	2	21	9	10	7	3	1	19	8	20	17	5	13	6	12	16	11	22	18	4	14	15	3596
5000	5	15	14	4	18	22	11	16	12	6	13	5	17	20	8	19	1	3	7	10	9	21	2	3596
5000	25	2	21	9	10	7	3	1	19	8	20	17	5	13	6	12	16	11	22	18	4	14	15	3596
5000	100	5	13	6	12	16	11	22	18	4	14	15	2	21	9	10	7	3	1	19	8	20	17	3596



Şekil 4.10: 22 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

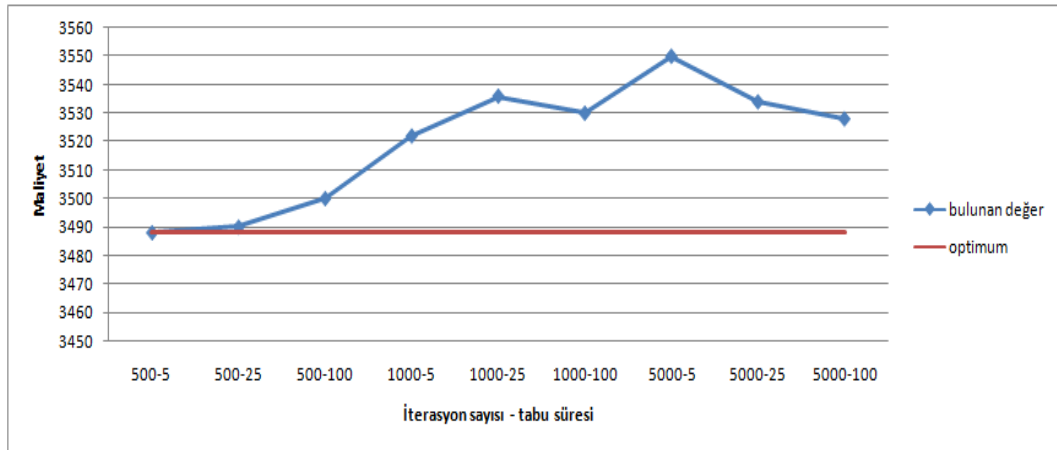
4.1.11. NUG24 Problemi için Elde Edilen Sonuçlar

NUG24 problemi 24 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.12’de, optimum sonuçlarla karşılaştırılması ise Şekil 4.11’de görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 3488 birimdir. Kullanılan algoritma ile bazı parametrelerde optimum değere farklı sıralamalar ile ulaşılabilirken diğer parametrelerde optimuma çok yakın değerler bulunmakta ve bu sonuçlar yaklaşık 780-1740 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.12: 24 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																					Maliyet				
500	5	17	8	11	23	4	20	15	19	22	18	3	14	1	10	7	9	16	21	24	12	6	13	5	2	3488
500	25	2	5	13	6	12	24	21	9	16	7	10	1	14	3	18	22	19	15	20	4	23	11	8	17	3490
500	100	14	20	3	21	2	5	4	23	18	16	9	13	11	8	22	7	10	6	15	17	19	1	24	12	3500
1000	5	5	24	1	18	23	17	6	12	10	22	11	15	13	9	7	19	8	14	2	21	16	3	4	20	3522
1000	25	4	3	11	23	18	14	21	16	7	19	8	20	2	9	10	22	1	17	5	13	6	12	24	15	3536
1000	100	15	6	22	11	16	4	17	1	18	23	8	19	24	12	10	7	3	14	5	2	13	9	21	20	3530
5000	5	2	21	9	13	6	24	5	16	7	10	12	1	20	3	8	11	22	15	14	4	19	23	18	17	3550
5000	25	2	5	21	3	14	20	13	6	16	18	23	4	9	12	7	22	11	15	10	24	1	19	8	17	3534
5000	100	20	3	19	21	2	5	14	18	8	22	6	15	11	23	7	1	12	24	4	16	9	10	13	17	3528



Şekil 4.2: 24 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

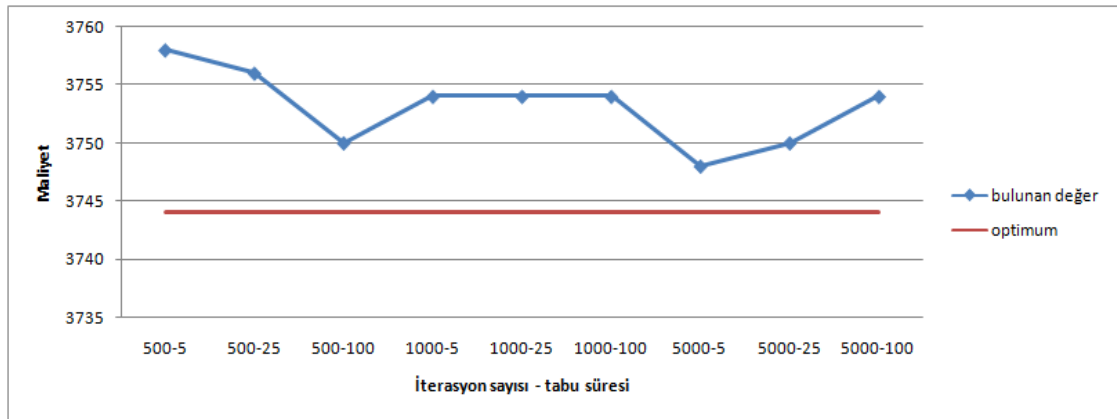
4.1.12. NUG25 Problemi için Elde Edilen Sonuçlar

NUG25 problemi 25 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.13'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.12'de görülmektedir.

Literatürde Clausen vd. tarafından bulunan en iyi maliyet 3744 birimdir. Kullanılan algoritma ile optimum değere ulaşılamasa da optimuma çok yakın değerler bulunmakta ve bu sonuçlar yaklaşık 870-1920 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.13: 25 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																									Maliyet	
500	5	13	4	23	17	12	24	7	14	3	18	21	19	6	16	11	1	10	9	25	2	15	20	22	8	5	3758
500	25	12	17	3	23	13	4	14	7	10	20	21	16	6	19	1	11	25	8	9	15	24	18	2	22	5	3756
500	100	17	24	18	2	5	12	3	16	25	11	4	14	6	8	21	23	7	19	9	20	13	10	1	22	15	3750
1000	5	5	2	11	20	15	18	25	8	9	22	3	16	6	19	1	24	21	14	7	10	17	12	4	23	13	3754
1000	25	5	2	11	20	15	18	25	8	9	22	3	16	6	19	1	24	21	14	7	10	17	12	4	23	13	3754
1000	100	13	10	1	22	15	23	7	19	9	20	4	14	6	8	11	12	21	16	25	2	17	24	3	18	5	3754
5000	5	17	12	4	23	13	24	21	14	7	10	18	16	3	6	19	2	25	8	9	1	5	11	20	22	15	3748
5000	25	5	2	18	24	17	11	25	16	3	12	21	8	6	14	4	20	9	10	7	23	15	22	1	19	13	3750
5000	100	22	20	11	2	5	15	9	8	25	18	1	21	6	16	24	19	10	7	3	17	13	23	14	4	12	3754



Şekil 4.12: 25 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

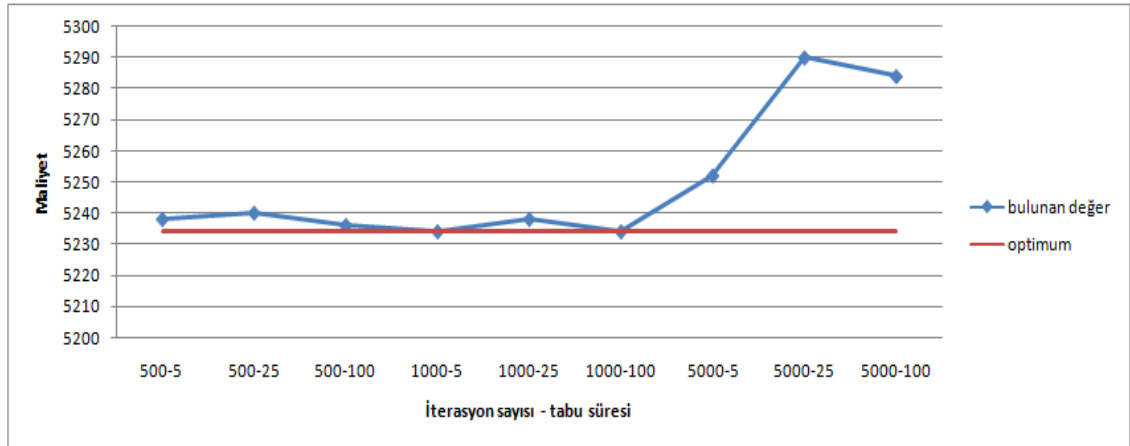
4.1.13. NUG27 Problemi için Elde Edilen Sonuçlar

NUG27 problemi 27 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.14'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.13'te görülmektedir.

Literatürde Anstreicher vd. (2001) tarafından bulunan en iyi maliyet 5234 birimdir. Kullanılan algoritma ile bazı parametrelerde optimum değere tek bir sıralama ile ulaşılabilirken diğer parametrelerde optimuma yakın değerler bulunabilmekte ve bu sonuçlar yaklaşık 1020-2220 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.14: 27 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																											Maliyet	
500	5	4	16	3	8	19	7	9	13	21	15	27	18	11	22	1	10	6	2	14	20	23	25	17	24	26	12	5	5238
500	25	14	15	18	23	17	1	24	12	5	20	27	11	8	22	10	26	6	2	4	16	3	25	19	7	9	13	21	5240
500	100	21	13	9	25	7	19	3	16	4	2	6	26	10	22	8	11	27	20	5	12	24	17	1	23	18	15	14	5236
1000	5	5	12	26	24	25	17	23	20	14	2	6	10	1	22	11	18	27	15	21	13	9	7	19	8	3	16	4	5234
1000	25	4	16	3	8	19	7	9	13	21	15	27	18	11	22	1	10	6	2	14	20	23	25	17	24	26	12	5	5238
1000	100	5	12	26	24	25	17	23	20	14	2	6	10	1	22	11	18	27	15	21	13	9	7	19	8	3	16	4	5234
5000	5	5	12	26	24	17	23	18	20	14	2	6	10	1	11	22	8	27	15	21	13	9	7	25	19	3	16	4	5252
5000	25	15	16	8	19	25	1	24	13	17	20	27	11	22	7	10	12	6	26	14	4	3	18	23	9	21	2	5	5290
5000	100	14	20	27	18	3	9	21	2	5	4	16	11	23	22	10	6	13	26	15	25	8	19	7	1	12	24	17	5284



Şekil 4.13: 27 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

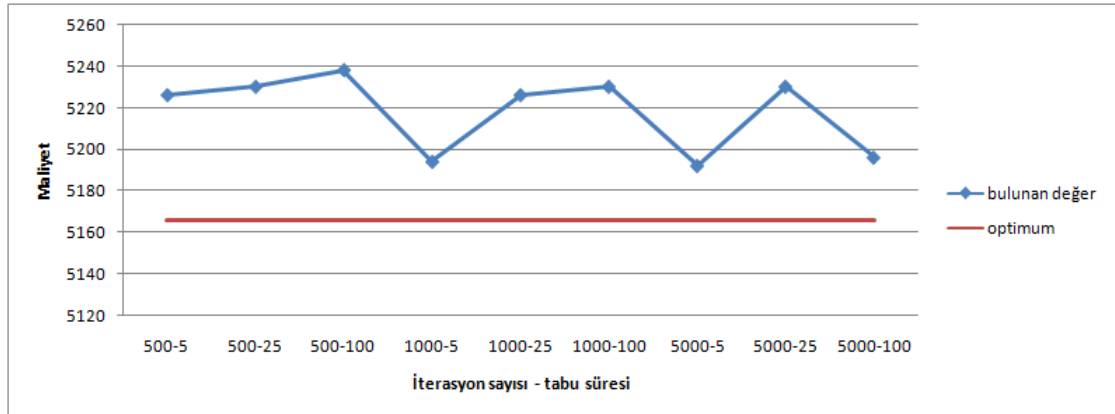
4.1.14. NUG28 Problemi için Elde Edilen Sonuçlar

NUG28 problemi 28 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.15'te, optimum sonuçlarla karşılaştırılması ise Şekil 4.14'te görülmektedir.

Literatürde Anstreicher vd. tarafından bulunan en iyi maliyet 5166 birimdir. Kullanılan algoritma ile optimum değere ulaşılamasa da optimuma yakın sayılabilecek değerler bulunmakta ve bu sonuçlar yaklaşık 1140-2520 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.15: 28 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																													Maliyet
500	5	15	17	23	26	24	12	5	14	20	18	22	1	6	2	27	8	19	10	7	13	21	4	16	3	11	9	25	28	5226
500	25	14	4	27	23	18	15	17	20	3	8	11	22	1	26	25	19	16	7	10	12	24	28	21	9	13	6	2	5	5230
500	100	20	28	21	6	13	2	5	14	16	3	7	9	12	24	27	8	19	22	18	10	1	15	4	25	11	23	26	17	5238
1000	5	5	26	24	17	11	23	4	2	6	12	1	22	18	15	13	9	10	7	3	27	14	21	28	25	19	8	16	20	5194
1000	25	5	2	9	11	23	3	4	15	12	6	22	18	27	14	26	1	10	7	8	16	20	17	24	13	25	19	21	28	5226
1000	100	28	25	19	3	21	2	5	4	16	8	7	9	13	17	20	27	11	22	10	6	26	14	15	23	18	1	12	24	5230
5000	5	14	15	23	18	12	24	17	20	27	11	22	1	10	26	4	3	8	19	7	9	5	28	16	25	6	21	13	2	5192
5000	25	21	28	16	19	3	20	14	2	25	7	8	11	27	4	13	6	9	10	22	18	15	5	12	24	1	26	23	17	5230
5000	100	5	26	24	12	17	23	15	2	6	10	1	22	18	27	13	9	16	7	11	3	14	21	28	25	19	8	4	20	5196



Şekil 4.14: 28 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

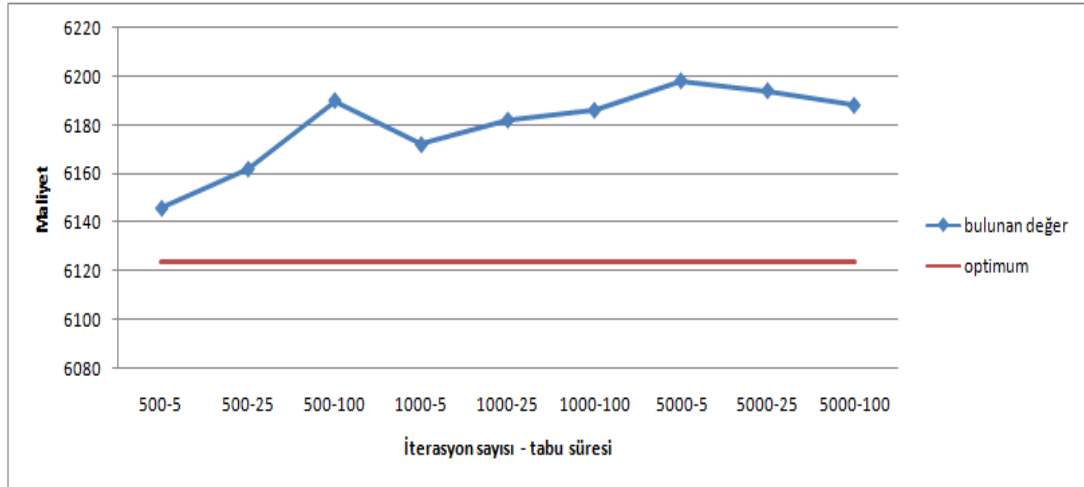
4.1.15. NUG30 Problemi için Elde Edilen Sonuçlar

NUG30 problemi 30 departmana sahip bir KAP örneğidir. Bu probleme ait iş akış ve uzaklık matrislerine <http://www.opt.math.tu-graz.ac.at/qaplib/inst.html> adresinden ulaşılabilir. Bu problemde 500, 1000 ve 5000 iterasyon sayılarının her biri için tabu süresi 5, 25 ve 100 olarak değiştirilerek çözümler elde edilmiştir. Elde edilen bu sonuçlar Tablo 4.16'da, optimum sonuçlarla karşılaştırılması ise Şekil 4.15'de görülmektedir.

Literatürde Anstreicher vd. tarafından bulunan en iyi maliyet 6124 birimdir. Kullanılan algoritma ile optimum değere ulaşılmasa da optimuma yakın sayılabilecek değerler bulunmakta ve bu sonuçlar yaklaşık 1440-2940 saniyelik bir sürede elde edilebilmektedir.

Tablo 4.16: 30 Departman için Bulunan Çözümler

İterasyon Sayısı	Tabu süresi																									Maliyet						
500	5	14	4	30	27	11	15	20	3	16	8	23	18	28	25	19	7	22	1	21	29	9	10	26	17	5	2	13	6	12	24	6146
500	25	15	11	27	30	4	14	18	23	8	16	3	20	1	22	7	19	25	28	24	12	10	9	29	21	17	26	6	13	2	5	6162
500	100	4	14	30	11	23	15	20	27	16	8	18	25	28	3	19	7	22	1	21	29	9	10	26	17	5	2	13	6	12	24	6190
1000	5	15	23	26	17	24	5	27	18	22	10	1	12	16	11	8	7	25	6	30	3	19	9	13	2	4	14	20	29	21	28	6172
1000	25	20	14	4	27	18	15	3	30	16	11	23	12	21	19	8	7	22	1	2	29	13	10	6	24	5	28	25	9	26	17	6182
1000	100	5	2	21	3	29	4	23	9	18	19	30	14	26	10	22	11	27	20	17	1	7	8	16	15	24	12	13	6	25	28	6186
5000	5	14	4	30	16	28	15	20	27	11	8	6	25	3	18	22	7	12	1	21	23	19	10	13	24	5	2	29	9	26	17	6198
5000	25	15	18	23	12	17	5	14	22	6	1	24	26	27	3	11	7	10	9	20	30	8	19	29	2	4	16	28	25	13	21	6194
5000	100	15	18	23	3	27	14	1	22	11	8	30	4	12	6	7	19	16	20	24	10	13	25	29	28	17	26	9	5	2	21	6188



Şekil 4.15: 30 Departman için Bulunan Çözümlerin Optimum Değerle Karşılaştırılması

4.1.16. Farklı Tabu Süreleri için Sonuçların Karşılaştırılması

Tabu süresinin boyutu için yapılan birçok çalışmaya rağmen hala kesin bir şey söylenememektedir. Tabu süresi çok kısa olursa aramanın yerel minimumda takılı kalma olasılığı vardır, tabu süresinin fazla uzun olması ise çözümün yasaklanan çözümlerin optimum çözüm olabilecek komşuluklarından da uzaklaşarak optimum değeri elde edemeyebilir.

Tablo 4.17’de görüldüğü gibi algoritma tabu süresi 5 iterasyon olduğu zaman 500 ve 5000 iterasyonluk aramalarda optimumu %66,67 oranında bulabilmekteyken, 1000 iterasyonluk aramalarda bu oran %60’a düşmüştür.

Tablo 4.17: Tabu süresi=5 için Elde Edilen En İyi Maliyet Değerleri

Tabu Süresi = 5															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
İterasyon Sayısı	500	500	500	500	500	500	500			500	500				500
	1000	1000	1000	1000	1000	1000			1000	1000			1000		
	5000	5000	5000	5000	5000	5000		5000		5000		5000		5000	
Maliyet	578	1014	1150	1610	1240	1732	1930	2570	2442	3596	3488	3748	5234	5192	6146
Optimum	578	1014	1150	1610	1240	1732	1930	2570	2438	3596	3488	3744	5234	5166	6124
Sapma	0	0	0	0	0	0	0	0	4	0	0	4	0	26	22

Tablo 4.18’de görüldüğü gibi tabu süresi 25 iterasyon olduğu zaman 500, 1000 ve 5000 iterasyonluk aramaların herbirinde optimum değer %66,67 oranında bulunmuştur.

Tablo 4.18: Tabu süresi=25 için Elde Edilen En İyi Maliyet Değerleri

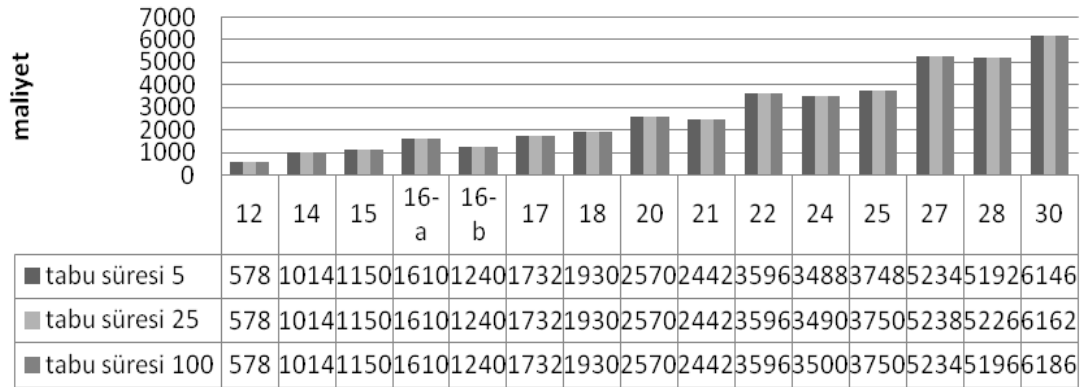
Tabu Süresi = 25															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
İterasyon Sayısı	500	500	500	500	500	500		500		500	500				500
	1000	1000	1000	1000	1000	1000	1000			1000			1000	1000	
	5000	5000	5000	5000	5000	5000		5000	5000	5000		5000			
Maliyet	578	1014	1150	1610	1240	1732	1930	2570	2442	3596	3490	3750	5238	5226	6162
Optimum	578	1014	1150	1610	1240	1732	1930	2570	2438	3596	3488	3744	5234	5166	6124
Sapma	0	0	0	0	0	0	0	0	4	0	2	6	4	60	38

Tablo 4.19’da görüldüğü gibi tabu süresi 100 iterasyon olduğu zaman 500 iterasyonluk aramalarda optimumu %73,33 oranında bulabilmekteyken, 1000 ve 5000 iterasyonluk aramalarda bu oran %60’a düşmüştür.

Tablo 4.19: Tabu süresi=100 için Elde Edilen En İyi Maliyet Değerleri

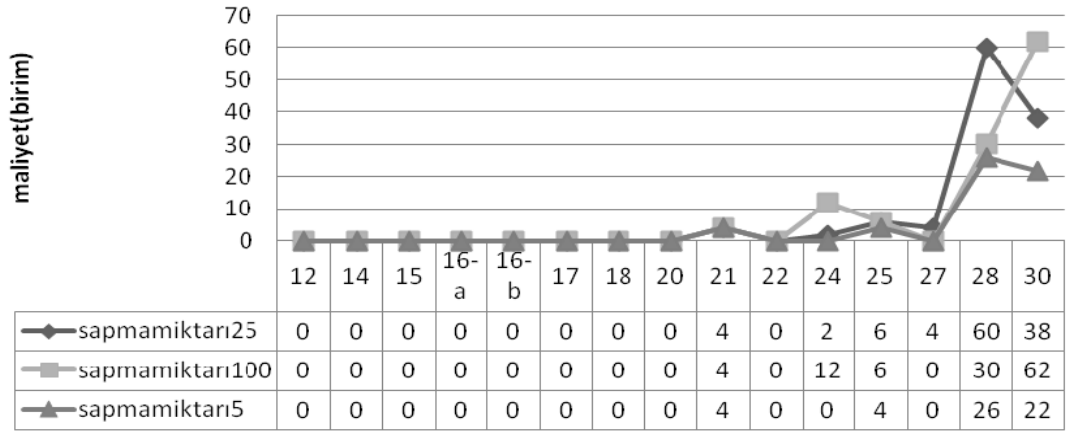
Tabu Süresi = 100															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
İterasyon Sayısı	500	500	500	500	500	500	500		500	500	500	500			
	1000	1000	1000	1000	1000	1000				1000			1000		1000
	5000	5000	5000	5000	5000	5000		5000		5000				5000	
Maliyet	578	1014	1150	1610	1240	1732	1930	2570	2442	3596	3500	3750	5234	5196	6186
Optimum	578	1014	1150	1610	1240	1732	1930	2570	2438	3596	3488	3744	5234	5166	6124
Sapma	0	0	0	0	0	0	0	0	4	0	12	6	0	30	62

Şekil 4.16'da tabu süresi 5, 25 ve 100 iterasyon olduğu durumlarda elde edilen en iyi maliyetler ve bunlar arasındaki karşılaştırmalar gösterilmiştir. 24, 25, 27, 28 ve 30 departmana sahip problemler dışındaki diğer problemlerde tabu süresinin uzunluğu optimum ya da optimuma yakın değerleri bulurken elde edilen maliyette bir fark oluşturmamıştır.



Şekil 4.16: Tabu Sürelerine Göre Maliyetlerin Karşılaştırılması

Şekil 4.17'de problemlerin 5, 25 ve 100 iterasyonlu tabu sürelerine sahip olduklarında en iyi maliyet değerlerinin literatürdeki optimum maliyet değerlerinden kaç birim sapma ile elde edildikleri gösterilmektedir. Grafiklere bakıldığında optimum ve optimuma en yakın değerler- 28 departmanlı problem haricinde- 5 iterasyonlu tabu sürelerinde elde edilmiştir. 100 iterasyonlu tabu sürelerinde ise -27 departmanlı problem haricinde- çoğu problemde optimum değere ulaşılmış olmasına rağmen, optimum maliyetten sapma miktarı en fazla olan değerler elde edilmiştir.



Şekil 4.37: Tabu Sürelerine Göre Maliyetten Sapma Miktarlarının Karşılaştırılması

4.1.17. Farklı İterasyon Sayısı için Sonuçların Karşılaştırılması

Tablo 4.20’de görüldüğü gibi algoritma iterasyon sayısı 500 olduğu zaman 5 ve 100 iterasyonlu tabu sürelerine sahip aramalarda optimumu %73,33 oranında bulabilmekteyken, 25 iterasyonlu tabu süresine sahip aramalarda bu oran %53,33’e düşmüştür.

Tablo 4.20: İterasyon Sayısı=500 için Elde Edilen En İyi Maliyet Değerleri

İterasyon Sayısı = 500															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
Tabu Süresi	5	5	5	5	5	5	5			5	5			5	5
	25	25	25	25	25	25		25		25					
	100	100	100	100	100	100	100		100	100		100	100		
Maliyet	578	1014	1150	1610	1240	1732	1930	2570	2442	3596	3488	3750	5236	5226	6146
Sapma	0	0	0	0	0	0	0	0	4	0	0	6	2	60	22

Tablo 4.21’de görüldüğü gibi algoritma iterasyon sayısı 1000 olduğu zaman 5 iterasyonlu tabu süresine sahip aramalarda %93,33 oranında optimum yaklaşılmışken, bu oran 25 ve 100 iterasyonlu tabu sürelerine sahip aramalarda %60’a düşmüştür.

Tablo 4.21: İterasyon Sayısı=1000 için Elde Edilen En İyi Maliyet Değerleri

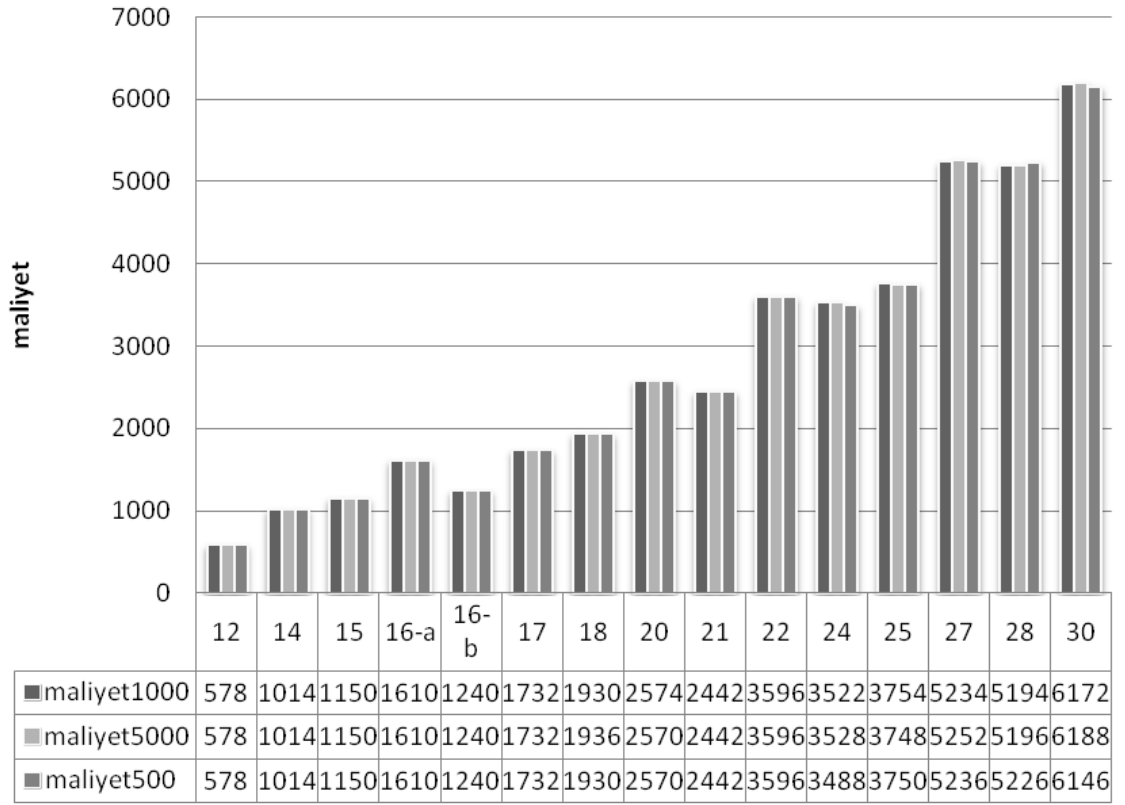
İterasyon Sayısı = 1000															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
İterasyon Sayısı	5	5	5	5	5	5		5	5	5	5	5	5	5	5
	25	25	25	25	25	25	25			25		25			
	100	100	100	100	100	100				100		100	100		
Maliyet	578	1014	1150	1610	1240	1732	1930	2574	2442	3596	3522	3754	5234	5194	6172
Sapma	0	0	0	0	0	0	0	4	4	0	34	10	0	28	48

Tablo 4.22’de görüldüğü gibi algoritma iterasyon sayısı 5000 olduğu zaman 5 ve 100 iterasyonlu tabu sürelerine sahip aramalarda optimum değeri elde etme oranı %73,33 iken bu oran 25 iterasyonlu aramalarda %60’a düşmüştür.

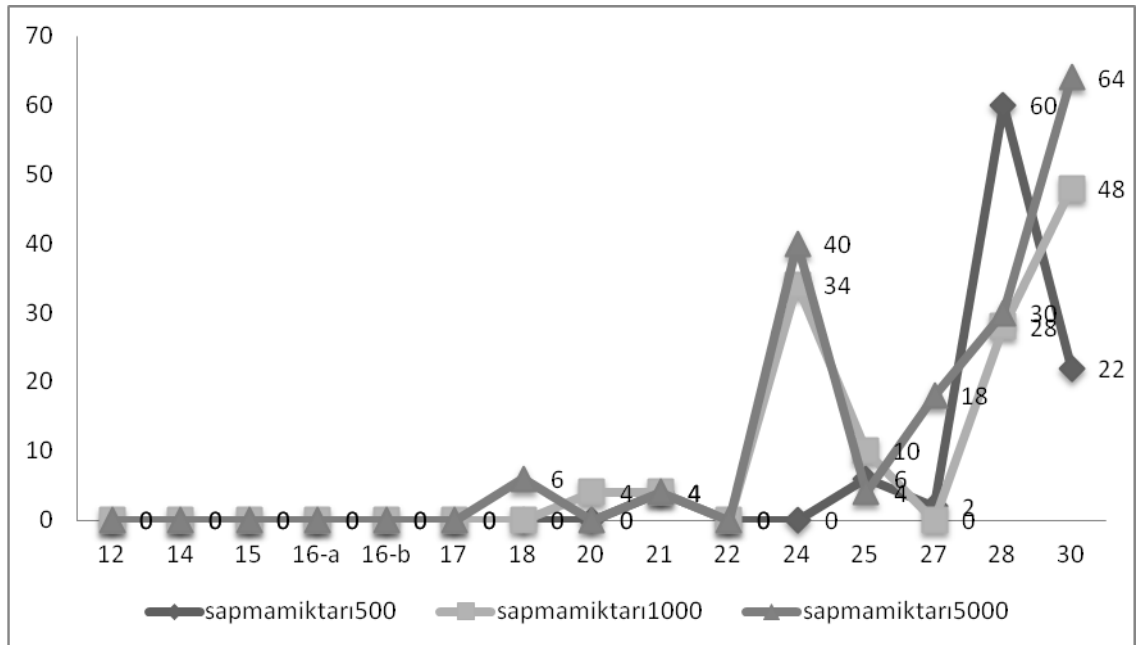
Tablo 4.22: İterasyon Sayısı=5000 için Elde Edilen En İyi Maliyet Değerleri

İterasyon Sayısı = 5000															
Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
İterasyon Sayısı	5	5	5	5	5	5		5		5		5	5	5	
	25	25	25	25	25	25		25	25	25					
	100	100	100	100	100	100	100	100		100	100				100
Maliyet	578	1014	1150	1610	1240	1732	1936	2570	2442	3596	3528	3748	5252	5192	6188
Sapma	0	0	0	0	0	0	6	0	4	0	40	4	18	30	64

Şekil 4.18’de iterasyon sayısı 500, 1000 ve 5000 olan durumlarda elde edilen en iyi maliyetler ve bunlar arasındaki karşılaştırmalar gösterilmiştir. 18, 20, 24, 25, 27, 28 ve 30 departmana sahip problemler dışındaki diğer problemlerde iterasyon sayısı optimum ya da optimuma yakın değerleri bulurken bir fark oluşturmamıştır.



Şekil 4.18: İterasyon Sayılarına Göre Maliyetlerin Karşılaştırılması



Şekil 4.19: İterasyon Sayılarına Göre Maliyetten Sapma Miktarlarının Karşılaştırılması

Şekil 4.19’da problemlerin 500, 1000 ve 5000 iterasyon sayılarına sahip olduklarında en iyi maliyet değerlerinin literatürdeki optimum maliyet değerlerinden kaç birim sapma ile elde edildiklerini göstermektedir. Grafıklere bakıldığında optimum ve optimuma en yakın değerler- 25, 27 ve 28 departmanlı problemler haricinde- 500 iterasyon sayısı ile elde edilmiştir. 5000 iterasyon sayısı ile -20, 25 ve 28 departmanlı problemler haricinde- çoğu problemde optimum değere ulaşılmış olmasına rağmen, optimum maliyetten sapma miktarı en fazla olan değerler elde edilmiştir. 1000 iterasyon sayısına sahip problemlerin çoğunda (11 tanesinde) optimum veya optimuma en yakın değerler elde edilirken, 20 ve 25 departmana sahip problemlerde küçük maliyet miktarlarıyla da olsa en fazla saptmaya sahip maliyet değerleri elde edilmiştir.

5. TARTIŞMA VE SONUÇ

Bu çalışmada optimum çözümleri elde etmek için son çözüme götüren adımın döngüsel hareketler yaratmasını engelleyen veya cezalandıran, sahip olduğu yapay hafıza sayesinde önceden incelenmiş çözümleri belirli bir süre üretmeyerek yerel optimumlardan kurtulabilen gelişmiş bir meta-sezgisel olan Tabu Arama yöntemi tasarlanmıştır. Bu TA algoritması ile literatürde KAP için sıkça kullanılan, C. E. Nugent, T. E. Vollmann ve J. Ruml (1968) tarafından yapılan çalışmalarla elde edilmiş, 12, 14, 15, 16, 17, 18, 20, 21, 22, 24, 25, 27, 28, 30 departman sayısına sahip, literatürde Tabu Arama Algoritması ile yapılmış, problem kümesinin çoğunun denendiği çok fazla çalışma bulunmayan NUG test problemleri üzerinde çalışılmıştır.

Bu çalışma ile elde edilen sonuçlara göre, tabu süresi ve iterasyon sayılarının algoritmanın optimum çözümleri bulma yeteneği üzerine etkisi incelenmiştir.

Algoritma başlangıç çözümünü rassal olarak belirlediği için birbirinden çok farklı sonuçlar da elde edilmiştir. Bu rassallık etkisini azaltmak için her problemde algoritma birden fazla kere denenmiştir. Optimum çözümleri elde etmek kadar bu çözümleri makul bir sürede elde edebilmek de önemli olduğu için deneme sayıları belirli bir süreyi aşmayacak şekilde (küçük problemlerde 5 dakika, büyük problemlerde ise 3 saat) kısıtlı tutulmuştur. 12-18 departmana sahip problemlerde 30'ar kez, 20-25 departmana sahip problemlerde 10'ar kez, 27-30 departmana sahip problemlerde ise 5'er kez deneme yapılmış ve elde edilen en iyi çözüm alınmıştır.

12, 14, 15, 16a, 16b, 17, 22 departmana sahip problemler ile yapılan çalışmaların sonucunda tüm parametrelerle optimum maliyetler elde edilebilmiştir.

18 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0-%0,004 oranında optimumdan sapma göstermiştir.

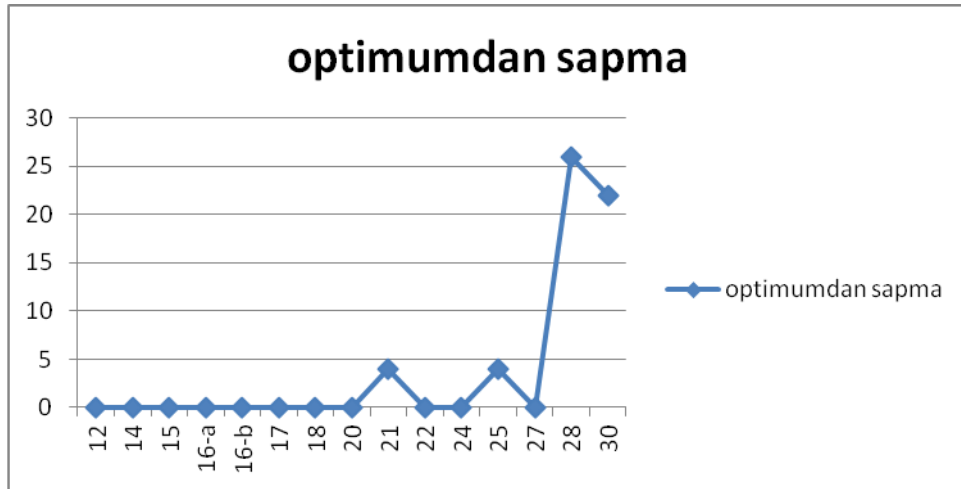
20 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0-%0,003 oranında optimumdan sapma göstermiştir.

Tablo 5.1 : NUG Problemleri için Literatürdeki En iyi Değerler ve TA ile Elde Edilen En İyi Maliyet Değerlerinin Karşılaştırılması

Departman Sayısı	12	14	15	16-a	16-b	17	18	20	21	22	24	25	27	28	30
Optimum	578	1014	1150	1610	1240	1732	1930	2570	2438	3596	3488	3744	5234	5166	6124
Maliyet	578	1014	1150	1610	1240	1732	1930	2570	2442	3596	3488	3748	5234	5192	6146
Sapma	0	0	0	0	0	0	0	0	4	0	0	4	0	26	22

21 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0,002-%0,004 oranında optimumdan sapma göstermiştir.

24 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0-%0,018 oranında optimumdan sapma göstermiştir.



Şekil 5.1 : Bulunan En İyi Maliyet Değerlerinin Optimum Değerlerden Sapma Miktarları

25 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0,001-%0,004 oranında optimumdan sapma göstermiştir.

27 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0-%0,011 oranında optimumdan sapma göstermiştir.

28 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0,005-%0,012 oranında optimumdan sapma göstermiştir.

30 departmana sahip problem ile yapılan çalışmanın sonucunda optimum maliyetten %0,004-%0,011 oranında optimumdan sapma göstermiştir.

İterasyon sayıları sabit tutularak tabu süresinin büyüklüğünün etkisi incelenmiştir. 500 iterasyon sayısına sahip bir çözümde 5 iterasyonlu tabu süresine sahip problemlerin elde edilen en iyi değerleri bulma oranı ile 100 iterasyonlu tabu süresine sahip problemlerden elde edilen en iyi değerlerin oranı aynı olsa da elde edilen sonuçlara bakılırsa 5 iterasyonlu tabu süresi daha iyi sonuçlar vermiştir.

1000 iterasyon sayısına sahip çözümlerde genel olarak optimum değeri verme oranı en yüksek olan 5 iterasyonlu tabu süresidir. 25 iterasyonlu tabu süresi ise 100 iterasyonlu tabu sürelerine göre daha iyi sonuçlar vermiştir.

İterasyon sayısı 5000 olarak sabit tutulduğunda 5 ve 100 iterasyonlu tabu sürelerine sahip sonuçların en iyi değerleri elde etme oranları 25 iterasyonlu tabu süresine sahip sonuçlara göre daha fazladır. Fakat genel olarak bakıldığında 5 iterasyonlu tabu süresi ile elde edilen sonuçların en iyi değerlere daha yakın olduğu görülmektedir.

İterasyon sayısının elde edilen sonuçlar üzerindeki etkisi istatistiksel olarak test edildiğinde elde edilen sonuçlara göre iterasyon sayıları arasındaki farkların anlamlı olmadığı, yani bu çalışmadaki verilere göre iterasyon sayısının elde edilen sonuçları çok fazla etkilemediği görülmektedir. İstatistiksel testlerin sonuçları tabloda görülmektedir ((a) Test istetistikleri, (b) Friedman Testi sonuçları, (c) Kendall's W Testi sonuçları).

Hipotezimiz ise şu şekildedir:

H_0 : Anakütle ortalamaları arasında anlamlı bir fark yoktur.

H_1 : Anakütle ortalamalarından en az biri anlamlı olarak diğerlerinden farklıdır.

Tablo 5.2: İterasyon Sayısına göre Test İstatistikleri

Test İstatistikleri					
İterasyon Süresi	N	Ort.	Std. Sapma	Min.	Max.
İS500	15	6,2667	15,92602	,00	60,00
İS1000	15	8,5333	15,31510	,00	48,00
İS5000	15	8,4000	17,58571	,00	64,00

(a)

Friedman Test İstatistikleri	
N	15
Chi-Square	,560
df	2
Asymp. Sig.	,756

(b)

Kendall's W Test İstatistikleri	
N	15
Kendall's W(a)	,019
Chi-Square	,560
df	2
Asymp. Sig.	,756

(c)

Tabu süresinin büyüklüğünü belirlemeye dair kesin bir yöntem olmasa da küçük değerlerin yerel optimuma takılmaya, büyük değerlerin ise arama esnasında optimum değerlerden uzaklaşmaya neden olabileceği bilinmektedir. Elde edilen sonuçlara bakıldığında tabu süresi 5 ve 100 iterasyon olduğunda en iyi sonuçları elde etme oranları yüksek olduğu için bu durumlar oluşmamış olabilir ya da başlangıç çözümünün etkisi ile aramaya optimum değerlere yakın komşuluktan başlanmış olabileceği için daha iyi sonuçlar bulunmuş olabilir. Tabu süresinin sonuçlar üzerindeki etkisini daha iyi anlayabilmek için tabu süreleri kaldırıldığında ya da tabu süresine iterasyon sayısı ile aynı değerler verildiğinde optimum değerleri elde etme oranları büyük oranda düşmüştür. Sonuç olarak, bu çalışmada elde edilen sonuçlara göre kullanılan tabu süresi değerlerinin iyi sonuçlar bulmak için uygun aralıkta olduğu söylenebilir.

Tabu süreleri sabit tutularak iterasyon sayılarının elde edilen sonuçlar üzerindeki etkisi incelenmiştir. Tabu süresi 5 iterasyon olarak sabit tutulduğunda 500 ve 5000 iterasyon sayılı çözümlerle 1000 iterasyonlu çözümlere göre çok az bir farkla daha iyi sonuçlar elde edilmiş olsa da iterasyon sayılarındaki değişimin çok etkili olmadığı görülmektedir.

Tabu süresi 25 iterasyon olarak sabit tutulduğunda ise iterasyon sayılarındaki değişim elde edilen sonuçlarda en iyi değeri bulma oranını etkilememiştir. Tabu süresi 100 iterasyon olduğunda ise 500 iterasyon sayılı çözümler elde edilen sonuçlara göre iterasyon sayılarından belirgin bir şekilde etkilenmiyor denilebilir, 1000 ve 5000 iterasyonlu çözümlere göre az bir farkla da olsa daha iyi değerler elde etmiş gibi

görünse de, bu durum başlangıç çözümünün rassal olmasından kaynaklanan bir durumdur.

Elde edilen sonuçlara bakıldığında 500 iterasyon sayısı ile de 1000 ve 5000 iterasyon sayısı ile bulunan iyi çözümler bulunabilmektedir. Bu durumda zaman ve maliyet kaybı olmaması için 500 iterasyonlu denemeler tercih edilebilir.

Tabu süresinin elde edilen sonuçlar üzerindeki etkisi istatistiksel olarak test edildiğinde elde edilen sonuçlara göre tabu süreleri arasında en az birinin diğerlerinden farklı olduğu, yani bu çalışmadaki verilere göre tabu süresi 5 iterasyon olan problemlerde diğerlerine göre daha iyi sonuçlar elde edildiği, tabu süresi 25 ve 100 iterasyon olan problemler arasında ise anlamlı bir fark olmadığı görülmektedir. İstatistiksel testlerin sonuçları tabloda görülmektedir ((a) Test istetistikleri, (b) Friedman Testi sonuçları, (c) Kendall's W Testi sonuçları).

Tablo 5.3: Tabu Süresine göre Test İstatistikleri

Test İstatistikleri					
Tabu Süresi	N	Ort.	Std. Sapma	Min.	Max.
tabusure5	15	3,7333	8,37911	,00	26,00
tabusure25	15	7,6000	17,42248	,00	60,00
tabusure100	15	7,6000	17,05788	,00	62,00

(a)

Friedman Test İstatistikleri	
N	15
Chi-Square	6,778
df	2
Asymp. Sig.	,034
N	15

(b)

Kendall's W Test İstatistikleri	
N	15
Kendall's W(a)	,226
Chi-Square	6,778
df	2
Asymp. Sig.	,034

(c)

Bu çalışmada, literatürde yaygın olduğu gibi başlangıç çözümü rastgele olarak oluşturulmuştur. Bu durum olumsuzluğu, yerel optimuma takılma ve istenen verimlilikte bir çözüme ulaşılamamasıdır. Son yıllarda yapılan çalışmalar literatürdeki mevcut sezgisellerden yararlanarak oluşturulan başlangıç çözümleri ile daha verimli çözümlerin elde edilmesinin mümkün olduğunu göstermektedir. Tabu Algoritmasının etkinliğini arttırmak için burada da çeşitli sezgiseller oluşturularak çözümün uygunluğu arttırılabilir ya da CORELAP programında olduğu gibi adım adım uygun bir çözüm elde edilerek bu uygun çözüm üzerinden algoritma aramaya başlayabilir. Ayrıca, bu çalışmada en fazla 30 departmanlı problemler ele alınmıştır. İleriki çalışmalarda daha fazla departmana sahip büyük problemlerde algoritmanın etkinliği denenebilir

KAYNAKLAR

- Akyıldız, G., Gürsoy, M., 2007, Havaalanı Kapı Ataması Problemine İlişkin Bir İnceleme, *Teknik ve Bilimsel Etkinlikler: Ulaştırma Kongresi 2007*, Eylül
- Aladağ, Ç. H., 2009, *Yapay Sinir Ağlarının Mimari Seçimi için Tabu Arama Algoritması*, Doktora Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü
- Alba, E., Leguizamo'n, G., Ordo'n ez. G., 2004, Parallel ant algorithms for the minimum tardy task problem, *Congreso Argentino de Ciencias de la Computacion*, San Justo, Argentina, October , vol. CACIC'04, 1835–1846.
- Aytekin, M. A., Kalaycı, T. E., 2010, Gezgin Satıcı Probleminin İkili Kodlanmış Genetik Algoritmalarla Çözümünde Yeni Bir Yaklaşım, *Akademik Bilişim 2010*
- Azadivar, F., Wang, J., 2000, Facility layout optimization using simulation and genetic algorithms, *International Journal of Production Research*, 38(17), 4369-4383.
- Balakrishnan, J., Cheng, C.H., Conway, D.G. and Lau, C.M., 2003, A hybrid genetic algorithm for the dynamic plant layout problem, *International Journal of Production Economics*, 86(2), 107-120.
- Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M., 2005, *Discrete - Event System Simulation*, Prentice-Hall, New Jersey, 485-492, ISBN: 9780131446793
- Baykasoğlu, A., Dereli, T., Sabuncu, İ., 2006, An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems, *The International Journal of Management of Science Omega*, 34, 385 – 396.
- Belgin, Ö., 2007, *Haberleşme Şebekelerinin Tasarımında Sezgisel Yaklaşımlar: Değişken Komşu Arama, Kuş Sürüsü Optimizasyonu, Karınca Kolonisi Optimizasyonu*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü
- Benjamin, C., O., Chi, S.-C., Gaber, T., Riordan, C., A., 1995, Comparing BP and ART II neural network classifiers for facility location, *Computers and Industrial Engineering*, 28(1), 43-50(8).
- Bhowmik, R., 2008, An Approach to the Facility Layout Design Optimization, *International Journal of Computer Science and Network Security*, 8 (4), 212-220

Cura, T., 2008, Doğrusal Olmayan Küresel Optimizasyon Problemleri için Tabu Arama Algoritmasının Kullanılması, *İ.Ü. İşletme Fakültesi Dergisi*

Cura, T., 2008, *Modern Sezgisel Teknikler ve Uygulamaları 1.Baskı*, Papatya Yayıncılık, İstanbul, ISBN: 978-975-6797-79-2

Değertekin, S. Ö., Ülker, M., Hayalioğlu, M. S., 2006, Uzay Çelik Çerçeveslerin Tabu Arama ve Genetik Algoritma Yöntemleriyle Optimum Tasarımı, *İMO Teknik Dergisi*, 3917-3934

Dolgui, A., Proth, J. M., 2010, *Supply Chain Engineering Useful Methods and Techniques*, Springer, ISBN: 978-1-84996-016-8

Dreo, J., Petrowski, A., Siarry, P., Taillard, E., 2006, *Metaheuristics for Hard Optimization Methods and Case Studies With 140 Figures*, Springer Berlin Heidelberg, New York, ISBN-13 978-3-540-23022-9

Erkut, H., Baskak, M., 2003, *Stratejiden Uygulamaya Tesis Tasarımı*, İrfan Yayıncılık, İstanbul, ISBN: 975-371-035-6

Fırat, M., Güngör, M., 2004, Askı Madde Konsantrasyonu ve Miktarının Yapay Sinir Ağları ile Belirlenmesi, *İMO Teknik Dergi*, Yazı 219, 3267-3282

Francis, R. L., White, J. A., 1974, *Facility Layout and Location*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey

Gambardella, L., M., Taillard, D., and Dorigo, M., 1999, Ant colonies for the QAP, *Journal of Operational Research Society*, 50, 167-176.

Glover, F., 1989a, Candidate List Strategies and Tabu Search, Technical Report, Graduate School of Business Administration, University of Colorado, Boulder

Glover, F., 1989b, Tabu search-Part I, *ORSA Journal on Computing*, 1(3), 190-206

Glover, F. 1990a. Tabu Search-Part II, *ORSA Journal on Computing*, 2, 4-32

Glover, F., 1990b, Tabu search: A tutorial, Technical Report, University of Colorado, Boulder

Glover, F., 1995, *Tabu search fundamentals and uses*, Technical Report, Graduate School of Business Administration, University of Colorado, Boulder

Glover, F., 1996, *Tabu search and adaptive memory programming – advances, applications and challenges*, Technical Report, College of Business, University of Colorado, Boulder

Glover, F., Laguna, M., 1993, Tabu Search, *Modern Heuristic Techniques for Combinatorial Problems*, C. Reeves, ed., Blackwell Scientific Publishing, 70-141

Glover, F. ve Laguna, M., 1997, *Tabu Search*, Kluwer, MA

Glover, F., Laguna, M., 2002, *Tabu search in Pardalos and Resende: Handbook of Applied Optimization*, Oxford Academic Press

Gözüpek, D., Genç, G., 2009, Hücresel Ağlarda Kanal Planlama Problemine Tabu Araması Yaklaşımı, *Akademik Bilişim 2009*, Şubat

Güden, H., Vakvak, B., Özkan, B. E., Altıparmak, F., Dengiz, B., 2005, Genel Amaçlı Arama Algoritmaları ile Benzetim Eniyilemesi En İyi Kanban Sayısının Bulunması, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü Endüstri Mühendisliği Dergisi*, 16 (1), 2-15

Gülsün, B., Tuzkaya, G., Duman, C., 2009, Genetik Algoritma ile Tesis Yerleşimi Tasarımı ve Bir Uygulama, *Doğuş Üniversitesi Dergisi*, 10 (1), 73-87

Hamzaçebi, C., Kutay, F., 2004, Yapay Sinir Ağları ile Türkiye Elektrik Enerjisi Tüketiminin 2010 Yılına kadar Tahmini, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü Endüstri Mühendisliği Dergisi*, 19 (3), 227-233

Halaç, O., 2001, *Kantitatif Karar Verme Teknikleri*, Alfa Basım Yayın Dağıtım, İstanbul, ISBN: 975-316-740-7

Hansen, P., Mladenovic, N., Urosevic, D., 2006, Variable neighborhood search and local branching. *Computers & Operations Research*, Volume 33, Issue 10, 3034-3045.

Heragu, S. S., 1997, *Facilities Design*. PWS Publishing Company, USA, ISBN (10): 053495183X

Heragu, S. S., 2006, *Facilities Design 2. Edition*, iUniverse Books, USA, ISBN (10): 0-595-35938-8

İşlier, A. A., 2000, Dinamik Yerleşim için, Dinamik Üretim, *1. Bölüm, Otomasyon, Kasım 2000*, 64-68

Kumar, S. A., Suresh, N., 2009, *Operations Management*, New Age International (P) Ltd. Publishers, New Delhi, ISBN (13): 978-81-224-2883-4

Küçükdeniz, T., 2009, *Sürü Zekası Optimizasyon Tekniği ve Tedarik Zinciri Yönetimi Tekniğinde Bir Uygulama*, Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü

Lee, K-Y., Han, S-N., Roh, M-I., 2003, An improved genetic algorithm for facility layout problems having inner structure walls and passages, *Computers & Operations Research*, 30, 117-138.

Lee, K-Y., Roh, M-I., Jeong, H-S., 1998, An improved genetic algorithm for multifloor facility layout problems having inner structure walls and passages, *Computers & Operations Research*, 32, 879-899.

- Li, H., Love, P. E. D., 1998, Site-Level Facilities Layout Using Genetic Algorithms, *Journal of Computing in Civil Engineering*, 12(4), 227-231
- Liu, H., Abraham, A., 2007, An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems, *Journal of Universal Computer Science*, 13(9), 1309-1331.
- Mak, K., L., Wong, Y., S., Chan, F., T., S., 1998, A genetic algorithm for facility layout problems, *Computer Integrated Manufacturing Systems*, 11(1-2), 113-127.
- McKendall, A., R., Shang, J., 2006, Hybrid ant systems for the dynamic facility layout problem, *Computers & Operations Research*, 33, 790–803.
- Meller, R. D., Gau, K. Y., 1996, The Facility Layout Problem: Recent and Emerging Trends and Perspectives, *Journal of Manufacturing Systems*, 15(5), 351-366
- Merz, P., Freisleben, B., 1999, A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem, Proceedings of the 1999 International Congress of Evolutionary Computation (CEC'99), IEEE Press, 2063-2070.
- Michel, L., Hentenryck, P. V., 2003, A simple tabu search for warehouse location, *European Journal of Operational Research* 576-591
- Mladenovic, N., Hansen, P., 1997, Variable neighborhood search. *Computers & Operations Research*, 24 (11), 1097-1100.
- Monticelli, A. J., Romero, R., Asada, E. N., 2008, *Fundamentals of Tabu Search, Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*, Wiley-IEEE Press, 101-123, ISBN: 978-0-470-22585-1
- Montreuil, B., Ouazzani, N., Brotherton, E., Nourelfath, M., 2004, Antzone layout metaheuristic: Coupling zone-based layout optimization, ant colony system and domain knowledge. Proceedings of the 8th International Material Handling Research Colloquium, Graz, Austria, 301-331.
- Murty, K.G., 1995, *Operations Research: Deterministic optimization models*, Prentice Hall, New Jersey, 581
- Nabiyev, V. V., 2003, *Yapay Zeka*, Seçkin Yayınevi, Ankara, ISBN: 975-347-985-9
- Önüt, S., Tuzkaya, U. R., Doğanç, B., 2007, A particle swarm optimization algorithm for the multiple-level warehouse layout design problem, *Computers & Industrial Engineering*, Kasım
- Panneerselvam, R., 2006, *Production and Operations Management*, PHI Learning Pvt. Ltd., New Delhi, ISBN: 81-203-2767-5
- Petry, J., Wijeysooriya, A., 2006, Tabu Search with Pickup and Delivery, *SRU Symposium for Research and Scholarship*, USA

Quarterman, L., Amundsen, A. E., Nelson, W., Tuttle, H., 1997, *Facilities and Workplace Design :An Illustrated Guide*, Engineering & Management Press, Norcross, ISBN (13): 978-0898061666

Reeves, C.R., 1993, *Modern Heuristic Techniques for Combinatorial Problems*, Halsted Press, New York

Rezazadeh, H., Ghazanfari, M., Saidi-Mehrabad, M., Jafar-Sadjadi, S., 2009, An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem, *Journal of Zhejiang University Science A*, 10(4), 520-529.

Sait, S.M., Youssef, H., 1999, *Iterative Computer Algorithms With Applications in Engineering*, Computer Society, California

Shouman, M. A., Nawara, G. M., Reyad, A. H., Kh El-darandaly, 2001, Facility Layout Problem and Intelligent Techniques: A Survey, *Proceedings of 7th International Conference on Production Engineering Design and Control*, Alexandria (Egypt)

Singh, S., P., Sharma, R., R., K., 2006, A Review of various approaches to Facility Layout Problems, *International Journal of Advanced Manufacturing Technology*, 30, 425-433.

Solnon, C., 2008, Combining two pheromone structures for solving the car sequencing problem with Ant Colony Optimization, *European Journal of Operational Research* 191, 1043–1055.

Söyler, H., Keskindürk, T., 2007, Karınca Kolonisi Algoritması ile Gezgin Satıcı Probleminin Çözümü, 8. *Türkiye Ekonometri ve İstatistik Kongresi 24-25 Mayıs 2007 – İnönü Üniversitesi, Malatya*

Şahin, R., 2008, Dinamik Tesis Düzenlenme Problemi için Bir Tavlama Benzetimi Sezgiseli, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 23(4), 863-870

Şahin, R., Türkbey, O., 2010, Çok Amaçlı Tesis Yerleşim Problemi için Yeni Bir Melez Sezgisel Algoritma, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 25(1), 119-130

Taha, H.A., 2000, *Yöneylem Araştırması (Çeviren: Ş.Alp Baray-Şakir Esnaf)*, 4. Basım, Literatür Yayıncılık, İstanbul, ISBN: 975-8431-06-4

Talbi, E. G., 2009 , *Metaheuristics- From Design to Implementation*, John Wiley & Sons Inc. , New York

Tanyaş, M., 2000, *Endüstri Mühendisliğine Giriş*, İstanbul, İrfan Yayıncılık, ISBN: 3000106100269

Tezer, T., 2009, *Toplama ve Dağıtım Zaman Pencere Arç Rotalama Problemi için Kesin Çözüm Yaklaşımı ve Örnek Uygulamalar*, Yüksek Lisans Tezi, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü

Toksarı, M., D., 2004, *Karınca algoritması ve tesis yerleşimi problemlerine uygulanması*, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi

Tompkins J.A., White, J.A., 1984, *Facilities Planning*, John Wiley&Sons Inc., New York

Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco J.M.A, 1996, *Facilities Planning*, John Wiley&Sons Inc., ISBN: 0-471-00252-6

Tsuchiya, K., Bharitkar, S. and Takefuji, Y., 1996, A neural network approach to facility layout problems, *European Journal of Operational Research*, 89(3), 556-563.

Ulutaş, :B., 2008, *Dinamik Yerleşim Probleminin Çözümü için Bir Klonal Seçim Algoritması ve Uygulamaları*, Doktora Tezi, Osmangazi Üniversitesi Fen Bilimleri Enstitüsü

Utkan, F., 2006, *Tesis Yerleşim Problemleri için Takım Zekası Tabanlı Bir Rassal Eniyileme Algoritması*, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü

Ünler, A., 2006, *Sezgisel Yöntemlerle k-harmonik Ortalama Veri Kümeleme Eniyilemesi*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü

Yigit, V., Türkbey, O., 2003, Tesis Yerleşim Problemlerine Sezgisel Metotlarla Yaklaşım, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, 18(4), 45-56

Zhang, C. Y., Li, P. G., Guan, Z. L., Rao, Y. Q., 2006, A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem, *Computers & Operations Research*, 34(11), 3229-3242

Zhang, H., Wang, J., Y., 2008, Particle Swarm Optimization for Construction Site Unequal-Area Layout, *Journal of Construction Engineering and Management*, 134 (9), 739-749.

www.opt.math.tu-graz.ac.at/qaplib/inst.html

ÖZGEÇMİŞ

Ayşe Nur Adıgüzel 1 Ocak 1985 yılında Adana'da doğmuştur. İlköğretim ve ortaöğretiminin ardından Adana Anadolu Lisesi'nden 2003 yılında mezun olmuştur. 2004 yılında girdiği İstanbul Üniversitesi Endüstri Mühendisliği Bölümü'nün lisans programını 2008 yılında tamamlamıştır. Aynı yıl yine İstanbul Üniversitesi Endüstri Mühendisliği Bölümü'nde yüksek lisans eğitimine başlamıştır. Adıgüzel, 2011 yılı Temmuz ayından beri İstanbul Üniversitesi Endüstri Mühendisliği Anabilim Dalı'nda araştırma görevlisi olarak çalışmaktadır.