BAKTERİYEL BESİN ARAMA OPTİMİZASYONU ALGORİTMASI İLE MONTAJ HATTI DENGELEME

Yakup ATASAGUN

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Ağustos, 2010

ÖZET

Yüksek Lisans Tezi

BAKTERİYEL BESİN ARAMA OPTİMİZASYONU ALGORİTMASI İLE MONTAJ HATTI DENGELEME

Yakup ATASAGUN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Yakup KARA

2010, 50 + ix Sayfa

Jüri: Doç. Dr. Yakup KARA

Prof. Dr. Ahmet PEKER

Yrd. Doç. Dr. Ahmet BABALIK

Montaj hattı dengeleme problemi, montaj hattında yerine getirilecek görevlerin, aralarındaki öncelik ilişkileri ihlal edilmeden, belirlenen bir çevrim zamanını aşmayacak ve belirli bir performans ölçütünü en iyileyecek şekilde istasyonlara atanması problemidir. Basit düz ve U-Tipi montaj hattı dengeleme problemlerinin çözümü için çok sayıda kesin, sezgisel ve metasezgisel yaklaşımlar geliştirilmiştir. Bu çalışmada basit düz ve U-Tipi montaj hattı dengeleme problemlerinin çözümü için son yıllarda araştırmacıların ilgisini çekmiş bir optimizasyon algoritması olan Bakteriyel Besin Arama Optimizasyonu Algoritması tabanlı bir metasezgisel çözüm yaklaşımı geliştirilmiş ve performansı değerlendirilmiştir. Çalışma kapsamında, tek modelli ve deterministik görev zamanlı, görev sayıları 7 ile 111 arasında değişen düz ve U-Tipi montaj hattı dengeleme problemleri istasyon sayısı minimize edilmeye çalışılarak çözülmüştür. Elde edilen sonuçlar, optimal istasyon sayıları, karınca kolonisi optimizasyonu algoritması sonuçları ve genetik algoritma sonuçları ile karşılaştırılmıştır. Önerilen algoritma ile düz ve U-tipi dengeleme göz önünde bulundurulan, toplam 128 test probleminden 123'ünde, saniyelerle ifade edilebilecek sürelerde optimal sonuçlara ulaşılmıştır.

Anahtar Kelimeler: Montaj Hattı Dengeleme, U-Tipi Montaj Hatları, Bakteriyel Besin Arama Optimizasyonu Algoritması, Metasezgiseller

ABSTRACT

Master Thesis

ASSEMBLY LINE BALANCING BY BACTERIAL FORAGING OPTIMIZATION ALGORITHM

Yakup ATASAGUN

Selçuk University

Graduate School of Natural and Applied Sciences

Industrial Engineering Branch

Supervisor: Assoc. Prof. Dr. Yakup KARA

2010, 50 + ix Pages

Jury: Assoc. Prof. Dr. Yakup KARA

Prof. Dr. Ahmet PEKER

Assist. Prof. Dr. Ahmet BABALIK

Assembly line balancing is the problem of assigning tasks to workstations by optimizing a performance measure while satisfying precedence relations between tasks and cycle time restrictions. Many exact, heuristic and metaheuristic approaches have been proposed for solving simple straight and U-shaped assembly line balancing problems. In this study, a quite new optimization algorithm Bacterial Foraging Optimization Algorithm based heuristic approach is proposed for solving simple straight and U-shaped assembly line balancing problems and performance of the proposed algorithm is evaluated. A well known data set taken from the literature in which the number of tasks varies between 7 and 111 is solved for single model straight and U-shaped assembly line balancing problems with deterministic task times. The results are compared with optimal solutions and the results of an ant colony based optimization algorithm and a genetic algorithm. The proposed algorithm found optimal solutions for 123 of 128 test problems in seconds.

Keywords: Assembly Line Balancing, U-Shaped Assembly Lines, Bacterial Foraging Optimization Algorithm, Metaheuristics

ÖNSÖZ

Günümüzde birçok sektörde yaygın olarak kullanılan montaj hatları, üretim süreçlerinin son aşamaları olmaları nedeniyle, işletmelerin genel performansı üzerinde oldukça etkilidir. Buna bağlı olarak işletmelerin karşı karşıya kaldığı önemli problemlerden biri, NP-Zor sınıfındaki montaj hattı dengeleme problemlerinin etkili bir şekilde çözülmesidir. Öte yandan, geçtiğimiz birkaç yıl içerisinde ortaya çıkmış ve yeni gelişmekte olan Bakteriyel Besin Arama Optimizasyonu Algoritması, montaj hattı dengeleme problemleri üzerinde henüz uygulanmamış bir optimizasyon algoritmasıdır. Bu tez çalışmasında, Bakteriyel Besin Arama Optimizasyonu Algoritması montaj hattı dengeleme problemlerine uygulanarak, montaj hattı dengeleme literatürüne yeni bir metasezgisel yöntem kazandırılmaya çalışılmıştır.

Yüksek lisans tez çalışmam boyunca, değerli katkılarını ve desteklerini esirgemeyen, son derece rahat ve huzurlu bir çalışma ortamı sağlayan Endüstri Mühendisliği Bölümü Başkanı Sayın Prof. Dr. Ahmet PEKER'e; yüksek lisansa başladığım günden bu güne her türlü fedakârlığı ve manevi desteği gösteren, montaj hattı dengeleme konusuna yönlendirerek beni yolundan gitmeye layık gören, bilgi birikimi ve tecrübesini büyük bir sabır ve anlayışla aktaran, hayatımda karşıma çıkan en büyük şans olarak gördüğüm ve alanında yetkin bir akademisyen olma yolunda, her adımda kendisini örnek aldığım danışmanım Sayın Doç. Dr. Yakup KARA'ya; yüksek lisans tezinin yoğun ve sıkıntılı süreçlerinde manevi desteklerini ve özellikle de anlayışlarını esirgemeyen mesai arkadaşlarıma ve bu günlere gelmemde tartışmasız çok büyük emeği olan aileme sonsuz teşekkürlerimi sunarım.

Yakup ATASAGUN

KISALTMALAR VE SİMGELER

Kısaltmalar:

BBAOA	: Bakteriyel besin arama optimizasyonu algoritması
GA	: Genetik algoritma
MHD	: Montaj hattı dengeleme
KKOA	: Karınca kolonisi optimizasyonu algoritması
TA	: Tabu araması
TB	: Tavlama benzetimi

Simgeler:

S	: Bakteri sayısı
i	: Bakteri indisi
N_k	: Kemotaktik süreç (kemotaksis) sayısı
$N_{ür}$	: Üreme sayısı
N_{ed}	: Eliminasyon-dağılma sayısı
P_{ed}	: Eliminasyon-dağılma olasılığı
$c(i)$	: Adım uzunluğu
N_s	: Adım sayısı sınırı
J	: Bir bakteriye ait çözümün maliyet
n	: Bir bakteriye ait çözümdeki toplam istasyon sayısı
C	: Çevrim zamanı
t_{son}	: Bir bakteriye ait çözümdeki son istasyonun iş yükü
$GAGK_k$	: Görev sırasının k. pozisyona görev eklemek üzere oluşturulmuş geçici atanabilir görevler kümesi
AGK_k	: Görev sırasının k. pozisyona görev eklemek üzere oluşturulmuş atanabilir görevler kümesi
SG	: Atanabilir görevler kümesinden seçilen görev
BZ	: İstasyon boş zamanı
$\varphi(i)$	: i bakterisi için belirlenen yön
$\Delta(i)$	: i bakterisi için [-1,1] aralığında rastsal olarak geliştirilmiş vektör

ŞEKİL LİSTESİ

Sayfa

Şekil 3. 1 Yedi görevli örnek bir öncelik diyagramı 12

Şekil 3. 2 Dengelenmiş bir düz montaj hattı 14

Şekil 3. 3 Dengelenmiş bir U-Tipi montaj hattı 17

Şekil 3. 4 BBAOA'nın akış diyagramı 25

Şekil 3. 5 Örnek bir çözüm kodlaması ve ilgili kodun çözümü 26

Şekil 3. 6 Örnek problem için bir bakteriye ait başlangıç çözümü 31

Şekil 3. 7 Örnek problem için kemotaksis sürecinde elde edilen çözüm 33

Şekil 4. 1 K. & Wester problemi için optimal sonucu veren bir bakteriye ait optimizasyon süreci 42

TABLO LİSTESİ

Sayfa

Tablo 3. 1 Merten (1967) problemi için görev süreleri.....29

Tablo 3. 2 Atama ağırlıkları için örnek başlangıç değerleri.....30

Tablo 3. 3 Örnek problem için elde edilen yön vektörleri 32

Tablo 3. 4 Örnek problem için güncellenmiş atama ağırlıkları.....32

Tablo 3. 5 20 görevli bir problem için örnek parametre seviyeleri 35

Tablo 4. 1 Önerilen BBAOA için düz MHD sonuçları.....36

Tablo 4. 2 Önerilen BBAOA için U-Tipi MHD sonuçları.....38

Tablo 4. 3 Düz ve U-Tipi MHD problemleri için BBAOA, KKOA ve GA karşılaştırması.....40

İÇİNDEKİLER

Sayfa

ÖZET ... iii
ABSTRACT ... iv
ÖNSÖZ ... v
KISALTMALAR VE SİMGELER ... vi
ŞEKİL LİSTESİ ... vii
TABLO LİSTESİ ... viii
İÇİNDEKİLER ... ix
1. GİRİŞ ... 1
2. KAYNAK ARAŞTIRMASI ... 4
3. MATERYAL VE METOT ... 9
3.1. Materyal ... 9
3.1.1. Montaj hattı dengeleme problemleri ... 9
3.1.1.1. Montaj hattı dengeleme problemi ile ilgili temel kavramlar ... 10
3.1.1.2. Düz montaj hattı dengeleme problemi ... 12
3.1.1.3. U-Tipi montaj hattı dengeleme problemi ... 14
3.1.2. Araştırmada kullanılan test problemleri ... 18
3.2. Metot ... 19
3.2.1. Bakteriyel besin arama optimizasyonu algoritması ... 19
3.2.1.1. Bakteriyel besin arama optimizasyonu algoritmasının biyolojik temelleri ... 19
3.2.1.2. Çözüm kodlama, görev atama prosedürü ve başlangıç çözümleri ... 26
3.2.1.3. Kemotaksis ... 28
3.2.1.4. Üreme ... 28
3.2.1.5. Eliminasyon ve dağılma ... 29
3.2.1.6. Açıklayıcı bir örnek ... 29
3.2.2. Deney Tasarımı ... 34
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA ... 36
5. SONUÇ VE ÖNERİLER ... 43
6. KAYNAKLAR ... 45

1. GİRİŞ

Müşteri tercihlerinde her geçen gün artan farklılıklar, pazardaki varlığını sürdürmek isteyen işletmeler için, farklı beklentilere cevap verebilecek nitelikte ürün çeşitliliğine ve farklı ürünleri eş zamanlı olarak üretebilme yeteneğine sahip olmayı kaçınılmaz hale getirmiştir. Böyle bir müşteri profiline hitap eden ve müşteri ihtiyaçlarını en hızlı, en kaliteli ve en düşük maliyetle karşılayabilmeyi hedefleyen bir işletme, etkili, verimli, bir o kadar da esnek bir üretim sistemine sahip olmalıdır. Üretilecek ürünün özelliklerine, çeşitliliğine ve miktarına bağlı olarak uygun bir üretim sisteminin kullanılması, işletmelerin pazarda rekabet avantajı elde etmelerindeki en önemli unsurlardan birisidir (Kara 2004).

Montaj hatları günümüzde, otomotiv, elektronik, beyaz eşya gibi birçok sektörde yaygın olarak kullanılmaktadır. Genellikle üretim süreçlerinin son aşamaları olan montaj hatlarının performansı, üretim süreçlerinin genel performansı üzerinde önemli etkiye sahiptir. Sürekli gelişim anlayışını benimseyen çok sayıda işletme, üretim süreçlerinin performansını artırmak amacıyla, montaj hatlarını kullanmakta, diğer üretim aşamaları ve işletme dışından tedarik edecekleri malzemelerin akış hızını son montaj hattının ihtiyacına göre ayarlamaktadırlar.

Bir *montaj hattı*, belirli sayıda ardışık iş istasyonunun birbirlerine bir malzeme taşıma sistemi ile bağlanmasıyla meydana gelir. Malzemeler, bu istasyonlar arasında sabit bir taşıma hızıyla hareket eder. Her istasyonda, ürünün tamamlanması için gerekli olan görevlerden bazıları gerçekleştirilir ve hattın sonuna gelindiğinde, ürün tamamlanmış olur. *Görev*, bir montaj hattında tamamlanması gereken toplam işin, işin niteliğine göre bölünebilecek en küçük parçasıdır. Bu görevler, aralarındaki öncelik ilişkileri göz önünde bulundurularak *istasyon* adı verilen yerlerde gerçekleştirilir. Montaj hattı üzerindeki bir istasyona, o istasyona atanan görevlerin tamamlanması için ayrılan zamana *çevrim zamanı* adı verilir. Çevrim zamanı, bir montaj hattı üzerindeki bütün istasyonlar için eşit olup, malzemelerin istasyonlar arasındaki sabit taşıma hızını belirleyen faktördür (Gökçen ve ark. 2006).

Montaj hatlarının tasarlanmasında ve talep değişikliklerine göre üretim hızının tekrar ayarlanmasında ortaya çıkan en önemli problem, montaj hattı dengeleme (MHD) problemidir. MHD problemi, montaj hattında yerine getirilecek görevlerin, öncelik ilişkileri ve çevrim zamanı gibi probleme özgü kısıtlar sağlanmak koşuluyla bir optimizasyon kriterine bağlı kalınarak istasyonlara atanması problemidir.

Montaj hatları, ürün çeşitliliğine göre iki temel sınıfa ayrılabilir. Tek çeşit ürünün üretildiği montaj hatları *Tek Modelli Montaj Hatları*; farklı ürünlerin veya aynı ürünün farklı modellerinin üretildiği montaj hatları ise, *Çok/Karışık Modelli Montaj Hatları* olarak adlandırılır. Hattın şekli açısından bakıldığında ise, montaj hatları temel olarak *Düz* ve *U-Tipi* olmak üzere ikiye ayrılır. Düz montaj hatlarının en önemli özelliği, bu hatların düz bir çizgi şeklinde yerleştirilmiş olmalarıdır. U-Tipi montaj hatlarını düz montaj hatlarından ayıran en önemli özellik, hattın şeklinin giriş ve çıkışı aynı yönde olacak şekilde, U şeklinde (⊃) olmasıdır. U-Tipi montaj hatlarının iki temel avantajı vardır. Birincisi; U-Tipi montaj hattının özel bir tür üretim hücresi olması dolayısıyla hatta çalışan işçiler arasındaki görülebilirliğin, bu sayede işbirliği ve iletişimin yüksek olması, diğeri ise; öncelik ilişkilerine bağlı olarak hem hattın başlangıç hem de bitiş kısmından görev atanabilme imkânı sağlaması nedeniyle ortaya çıkan dengeleme esnekliğidir (Kara 2004).

1950'li yıllarda, verimli, etkili, esnek ve sürekli gelişim modelinin öncüsü olan bir üretim sistemi olarak Toyota Üretim Sistemin ortaya çıkmasıyla, verimliliği artırmak, sürekli gelişim anlayışını benimsemek ve kaliteden ödün vermeden maliyetleri düşürmek amacıyla uygulanması gereken tekniklerin hemen hepsi, tesis yerleşiminin U-Tipi hatlarla oluşturulmasını gerektirmektedir. Buna bağlı olarak, üretim tesislerini U-Tipi hatlarla tasarlayan işletmelerin karşısına çıkabilecek en önemli problem, U-Tipi hatların etkili bir şekilde dengelenmesi problemidir.

Basit düz ve U-Tipi MHD problemlerinin çözümü için çok sayıda kesin (exact), sezgisel (heuristic) ve metasezgisel (metaheuristic) yaklaşımlar geliştirilmiştir. Tavlama Benzetimi (TB), Tabu Araması (TA), Genetik Algoritmalar (GA) ve Karınca Kolonisi Optimizasyon Algoritması (KKOA) MHD problemlerinin

çözümünde yaygın olarak kullanılan metasezgisellerdir. Son yıllarda araştırmacıların ilgisini çekmiş bir metasezgisel yaklaşım olan Bakteriyel Besin Arama Optimizasyon Algoritması (BBAOA), bu problemlerin çözümünde henüz kullanılmamış ve bu problemlerin çözümündeki başarısı hakkında literatürde herhangi bir bilgi bulunmamaktadır. Bu çalışmanın amacı, basit düz ve U-Tipi MHD problemlerinin çözümü için BBAOA tabanlı bir çözüm yaklaşımı geliştirilmesi ve performansının değerlendirilmesidir. Bu amaç doğrultusunda, BBAOA tabanlı bir algoritma önerilmiş ve önerilen algoritmanın çözüm başarısı literatürde yaygın olarak kullanılan çok sayıda test problemi kullanılarak gerçekleştirilen deneyler ile değerlendirilmiştir. Çalışma kapsamında, tek modelli ve deterministik görev süreli düz ve U-Tipi MHD problemlerinin çözümü ele alınmıştır. Diğer MHD problemleri çalışma kapsamında ele alınmamıştır.

2. KAYNAK ARAŞTIRMASI

Tek modelli deterministik görev zamanlı düz MHD problemi ilk olarak Salveson (1955) tarafında ortaya atılmıştır ve basit MHD problemi olarak bilinmektedir. MHD problemlerinin en basit şekli olan tek modelli düz MHD problemi bu güne kadar birçok araştırmacının ilgisini çekmiştir. Konu ile ilgili literatür oldukça geniş olup Baybars (1986), Ghosh ve Gagnon (1989), Erel ve Sarin (1998) ve Becker ve Scholl (2006) tarafından ortaya koyulmuş literatür inceleme çalışmaları mevcuttur.

Tek modelli deterministik görev zamanlı U-Tipi MHD problemi ile ilgili ilk çalışma Miltenburg ve Wijingaard (1994) tarafından yapılmıştır. Çalışmada, problemin çözümü için bir dinamik programlama yaklaşımı önerilmiştir.

Hwang ve ark. (1996), işçilerin yürüme zamanlarını da dikkate alarak tek modelli deterministik görev zamanlı U-Tipi MHD problemi için bir tavlama benzetimi algoritması önermişlerdir.

Sparling (1997), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için optimal sonuçlar veren üç yeni çözüm yaklaşımı geliştirmiştir. Bu yaklaşımlardan biri dinamik programlama yaklaşımı, diğer ikisi ise dal-sınır algoritması yaklaşımlarıdır.

Deshpande ve ark. (1997), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için bir tamsayılı programlama yaklaşımı geliştirmişler ve çözümü için Langrange gevşetme yöntemini kullanmışlardır.

Urban (1998), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için bir tamsayılı programlama yaklaşımı önermiş ve elde ettiği sonuçları maksimum pozisyon ağırlığı metodunun sonuçları ile karşılaştırmıştır. Yaklaşımın testinde, görev sayısı 21 ile 45 arasında değişen toplam 25 adet problem kullanmıştır. Sonuçlar; tamsayılı programlama yaklaşımının bu 25 problemden 24'ü için optimal

sonuçları kabul edilebilir bir sürede verdiğini, bir adedi için ise çözüm süresinin bir saati aştığını göstermiştir.

Ajenblit ve Wainwright (1998), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için bir genetik algoritma yaklaşımı önermişler ve bu yaklaşımın performansını toplam 61 farklı problemden oluşan 12 problem setinde test etmişlerdir. Genetik algoritma, 61 problemin 21'inde optimal sonuçlara ulaşmış, diğer problemlerde ise karşılaştırma yapılan diğer sezgisel yaklaşımlara göre daha iyi sonuçlar vermiştir.

Aase (1999), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için farklı dallandırma stratejilerine sahip üç farklı dal-sınır algoritması yaklaşımı önermiştir.

Scholl ve Klein (1999), düz MHD problemi için geliştirdikleri ve SALOME adını verdikleri dal-sınır algoritması yaklaşımını, tek modelli deterministik görev zamanlı U-Tipi MHD problemi için düzenlemişler ve bu yaklaşıma da ULINO adını vermişlerdir.

Miltenburg (2000), U-tipi hatlarda arıza ortaya çıkması halinde U-tipi yerleşimin hat yönetimi üzerindeki etkisini araştırmıştır. Elde edilen çalışma sonuçları ile arıza durumunda U-tipi hat yerleşiminin düz hat yerleşimine göre daha etkin hat yönetimine sahip olduğunu belirtmiştir.

Cheng ve ark. (2000) geleneksel düz montaj hattı ve U-Tipi montaj hattı yerleşimlerini ürün kalitesi üzerindeki etkileri açısından karşılaştırmışlardır. Elde edilen sonuçlar, dikkate alınan bütün kalite karakteristikleri bakımından U-Tipi montaj hatlarının geleneksel montaj hatlarına karşı üstün olduğunu ortaya koymaktadır.

Erel ve ark. (2001), tek modelli deterministik görev zamanlı U-Tipi MHD problemi için tavlama benzetimi tabanlı bir sezgisel yaklaşım geliştirmişlerdir.

Ağpak ve Gökçen (2001), Arcus tarafından düz montaj hatları için geliştirilmiş olan COMSOAL metodunu tek modelli deterministik görev zamanlı U-Tipi MHD problemi için düzenlemişlerdir.

Miltenburg (2001a), U-tipi üretim ve montaj hatları ile ilgili teorik ve uygulamaya yönelik genel bir değerlendirme çalışması yapmıştır. Dokuz Amerikan ve 105 Japon işletmesini kapsayan toplam 114 U-tipi üretim tesisinde yaptığı analizler sonucunda şu sonuçlar ortaya çıkmıştır: U-tipi montaj hattındaki görev sayısının 1–24 aralığında değiştiği ve ortalamasının 10,2 olduğu, U-tipi montaj hatlarındaki istasyon sayısının 1–24 aralığında değiştiği ve ortalamasının 3,4 olduğu ve U-tipi montaj hatlarının %75'nin karma modelli üretim yaptığı belirlenmiştir. Çalışmada, Tam Zamanında Üretim felsefesi ile ortaya çıkan U-tipi montaj hatlarının işletmelerde üretimin etkinliğini gösterecek veriler belirtilmiştir. Buna göre U-tipi montaj hattında üretim yapılması sonucunda ortalama %76 verimlilik artışı, süreç içi stoklarda % 86 düşüş, üretim zamanlarında % 75 azalma ve hata oranlarında % 83 azalma tespit edilmiştir.

Miltenburg (2001b), U-tipi hatlarda tek parça akışı üretim üzerinde durduğu çalışmasında, konu ile ilgili literatürde yer alan çalışmaları değerlendirmiş ve yeni çalışma alanlarını belirlemiştir.

Ağpak ve Gökçen (2002), U-tipi montaj hattı literatüründe ilk olarak bulanık kavramını ele alarak, basit U-tipi MHD problemleri için bir bulanık tamsayılı programlama modeli geliştirmişlerdir.

Aase ve ark. (2003), tek modelli deterministik U-Tipi MHD problemi için geliştirilmiş dal–sınır algoritması yaklaşımlarından oluşan optimizasyon algoritmalarının değerlendirmesini yapmışlardır.

Aase ve ark. (2004), U-tipi montaj hatlarında iş gücü üretkenliğinin gelişimini deneysel olarak belirlemeye çalışmışlardır. Çalışmada elde edilen sonuçlara göre,

belirli şartlar altında düz montaj hattı yapısının U-tipi hat yapısına dönüştürülmesi ile işgücü üretkenliğinde önemli derecede gelişme sağlanabileceği belirtilmiştir.

Gökçen ve Ağpak (2006), tek modelli deterministik U-Tipi MHD problemi için bir hedef programlama yaklaşımı önermişlerdir.

Hwang ve ark. (2008) tarafından tek modelli deterministik U-Tipi MHD problemlerinin çözümü için çok amaçlı bir genetik algoritma yaklaşımı önerilmiştir.

Suwannarongsri ve Puangdownreong (2008), tek modelli deterministik U-Tipi MHD problemi için bir tabu arama (TA) algoritması geliştirmişlerdir. Literatürde bulunan üç farklı test probleminin çözümünde kullanılan tabu arama algoritmasının optimal çözümler verdiği belirtilmiştir.

Kara ve ark. (2009), tek modelli düz ve U-tipi MHD problemlerinin çözümü bulanık çok amaçlı programlama yaklaşımını kullanmışlardır. Açıklayıcı bir örnek üzerinde bulanık çok amaçlı yöntemin düz ve U-tipi MHD problemleri üzerindeki performanslarının karşılaştırmasını yapmışlardır.

Sabuncuoğlu ve ark. (2009) tarafından tek modelli deterministik U-Tipi MHD problemi için; Baykasoğlu ve Dereli (2009) tarafından ise basit düz ve U-Tipi MHD problemleri için Karınca Kolonisi Optimizasyonu Algoritmaları (KKOA) önerilmiştir.

Hwang ve Katayama (2010) tarafından tek modelli deterministik U-Tipi MHD problemlerinde düzgün iş yükü atamaları için iyileştirme yapısına sahip Genetik Algoritma tabanlı bir yaklaşım öne sürülmüştür.

Karışık modelli U-tipi MHD problemine yönelik olarak literatürdeki çalışmalar şunlardır: Sparling ve Miltenburg (1998), Kim ve ark. (2000), Miltenburg (2002), Kara ve ark. (2007a), Kara ve ark. (2007b), Kara (2008), Kara ve Tekin (2009).

MHD problemlerinin özel bir versiyonu olan, çok sayıda U-Tipi hattın bütünleşik olarak dengelenmesi problemi ilgili literatür oldukça kısıtlıdır. Miltenburg (1998) tarafından bir dinamik programlama yaklaşımı, Sparling (1998) tarafından ise bir sezgisel yöntem önerilmiştir. Chiang ve ark. (2007)'nın çalışması ise konsept olarak diğerlerinden farklı olup, aynı ürünü üretmek üzere, toplam çıktıları talebi karşılayacak şekilde tasarlanmış çok sayıda U-Tipi hattın dengelenmesine yöneliktir.

Karışık modelli çok sayıda U-Tipi hattın bütünleşik dengelenmesine yönelik olarak ise Sparling ve Miltenburg (1998)'a ait bir çalışma bulunmaktadır.

Bakterilerin besin arama stratejilerinden yola çıkılarak geliştirilen BBAOA son yıllarda mühendislik problemlerinin çözümünde kullanılmaya başlayan ve dikkat çeken metasezgisellerden biridir. Bremermann (1974), bakterilerin zengin besin alanlarını arama yolunun optimizasyon süreci olarak değerlendirilebileceğini belirtmiştir. Bakterilerin besin arama stratejilerini algoritma olarak ilk uygulayan Passino (2002)'dur. Passino (2002) çalışmasında bakterilerin besin arama davranışlarının biyolojik özelliklerini açıkladıktan sonra dağıtım optimizasyonu ve kontrolüne yönelik olarak BBAOA'yı uygulamıştır.

BBAOA, bugüne kadar optimal kontrol (Kim ve Cho 2005a), harmonik tahmin (Mishra 2005), makine öğrenmesi (Kim ve Cho 2005b), tahmin (Majhi ve ark. 2007), iletim kaybının azaltılması (Mishra ve ark. 2007); (Tripathy ve ark 2006), doğrusal olmayan sistemlerin tanımlanması (Majhi ve Panda 2007), çizelgeleme (Wu ve ark. 2007) gibi mühendislik problemlerinde başarıyla uygulanmıştır. BBAOA, diğer metasezgisel yaklaşımlar ile birleştirilerek de optimizasyon problemlerinin çözümünde kullanılmıştır. Kim ve ark. (2007) genetik algoritma ile BBAOA'yı, Biswas ve ark. (2007a) parçacık sürüsü optimizasyonu ile BBAOA'yı, Biswas ve ark. (2007b) BBAOA ile diferansiyel evrim algoritmasını birleştirerek melez yöntemler uygulamışlardır.

3. MATERYAL VE METOT

3.1. Materyal

3.1.1. Montaj hattı dengeleme problemleri

Montaj hatlarının tasarlanmasında ve talep değişikliklerine göre üretim hızının tekrar ayarlanmasında ortaya çıkan en önemli problem, MHD problemidir. *MHD problemi*, montaj hattında yerine getirilecek görevlerin, aralarındaki öncelik ilişkileri ihlal edilmeden, belirlenen bir çevrim zamanını aşmayacak ve belirli bir performans ölçütünü en iyileyecek şekilde istasyonlara atanması problemidir (Baybars 1986). Bu performans ölçütü genellikle, montaj hattında açılacak olan istasyon sayısı olmakla birlikte problemin amacı, bir istasyonda bir işçinin çalışacağını varsayarak ihtiyaç duyulacak işçi (istasyon) sayısını en aza indirmektir. MHD problemleri, ürün çeşitliliği, hattın şekli, görev zamanlarının durumu ve problemin performans ölçütüne göre farklı açılardan sınıflandırılabilmektedir.

Aşağıda MHD problemlerinin bir sınıflandırması verilmiştir.

Ürün çeşitliliğine göre;

- *Tek Modelli MHD Problemi,* tek çeşit ürün üretilen montaj hatlarının dengelenmesi problemidir.
- *Çok / Karışık Modelli MHD Problemi,* farklı ürünlerin veya aynı ürünün farklı modellerinin üretildiği montaj hatlarının dengelenmesi problemidir.

Hattın şekline göre;

- *Düz MHD Problemi,* düz bir çizgi şeklinde yerleştirilmiş montaj hatlarının dengelenmesi problemidir.
- *U-Tipi MHD Problemi,* U şeklinde yerleştirilmiş montaj hatlarının dengelenmesi problemidir.

Görev zamanlarının durumuna göre;

- *Deterministik Görev Zamanlı MHD Problemi,* görev zamanları belirli ve deterministiktir.
- *Stokastik Görev Zamanlı MHD Problemi,* görev zamanlarının, ortalaması ve standart sapması bilinen bir dağılıma uyduğu kabul edilir.

Performans ölçütüne göre;

- *Tip-I MHD Problemi,* problemin amacı, belirli bir çevrim zamanı için hat boyunca açılan istasyon sayısını minimize etmektir.
- *Tip-II MHD Problemi,* amacı belirli bir istasyon sayısı için hattın çevrim zamanını minimize etmektir.

Bu çalışma kapsamında, tek modelli, deterministik görev zamanlı, Tip-I, düz ve U-Tipi MHD problemleri ele alınmıştır.

3.1.1.1. Montaj hattı dengeleme problemi ile ilgili temel kavramlar

MHD problemlerinde kullanılan temel kavramlar ve tanımlar aşağıda verilmiştir.

***Montaj Hattı*:** Bir montaj hattı, birbirlerine bir malzeme taşıma sistemi ile bağlanmış, belirli sayıda ardışık istasyondan meydana gelir. Malzemeler, bu istasyonlar arasında sabit bir taşıma hızıyla hareket ettirilir. Her istasyonda, ürünün işlemlerinin tamamlanması için gerekli olan görevlerin bir kısmı tamamlanır ve hattın sonuna gelindiğinde, ürünün bütün işlemleri tamamlanmış olur.

***Görev*:** Bir montaj hattında tamamlanması gereken toplam işin, işin niteliğine göre bölünebilecek en küçük parçasıdır.

***Görev Zamanı*:** Bir görevin tamamlanabilmesi için gereken zamandır.

***İş İstasyonu*:** Bir montaj hattındaki görevler, aralarındaki öncelik ilişkileri göz önünde bulundurularak iş istasyonu adı verilen yerlerde gerçekleştirilir.

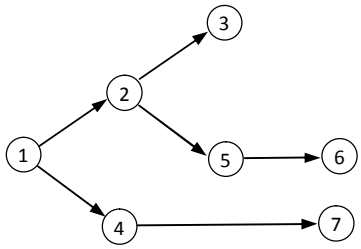
***İstasyon Zamanı*:** Montaj hattındaki bir istasyonda gerçekleştirilen görevlerin görev zamanlarının toplamıdır. Bir istasyonun istasyon zamanı, hattın çevrim zamanından küçük veya eşit olmak zorundadır.

***Çevrim Zamanı*:** Montaj hattını üzerindeki bir istasyona, o istasyona atanan görevlerin tamamlanabilmesi için ayrılan zamandır. Çevrim zamanı bir montaj hattını oluşturan bütün istasyonlar için eşittir ve hattan ardı ardına iki ürün çıkışı arasında geçen süreyi ifade eder. Montaj hattındaki görev zamanlarının en büyüğü, çevrim zamanı için alt sınırdır. Belirli bir zaman dilimindeki talebi karşılayacak şekilde bir ürünün üretilmesi için gereken zaman ise çevrim zamanı için üst sınırı oluşturur.

***İstasyon Boş Zamanı*:** Montaj hattının çevrim zamanı ile hatta açılan bir istasyonun istasyon zamanı arasındaki farktır.

Toplam Boş Zaman: Montaj hattını oluşturan bütün istasyonlarının boş zamanlarının toplamıdır.

***Öncelik Diyagramı*:** Bir montaj hattındaki görevler arasında bazı öncelik ilişkileri olabilir. Öncelik diyagramı, bu öncelik ilişkilerini ifade eder. Şekil 3.1'de yedi görevli örnek bir öncelik diyagramı verilmiştir. Daire içindeki sayılar görev numaralarını ifade etmektedir. Şekildeki 1 ve 2 numaralı görevleri ele alırsak, 1 numaralı görev, 2 numaralı görevden önce tamamlanmak zorundadır. Bu görevler arasındaki böyle bir ilişkiye *öncelik ilişkisi* adı verilir ve 1 numaralı görev, 2'nin öncülü; 2 numaralı görev ise 1'in ardılıdır denir.



Şekil 3. 1 Yedi görevli örnek bir öncelik diyagramı (Merten 1967)

3.1.1.2. Düz montaj hattı dengeleme problemi

Bu bölümde, MHD problemlerinin en yalın hali olan tek modelli deterministik görev zamanlı düz MHD probleminin temel varsayımları, kısıtları ve tamsayılı programlama modeli verilmiştir.

Tek modelli deterministik görev zamanlı düz MHD problemi ilk olarak Salveson (1955) tarafında ortaya koyulmuştur ve *basit MHD problemi* olarak bilinmektedir.

Problemin temel varsayımları şunlardır:

- Montaj hattında tek çeşit ürün büyük miktarlarda üretilir.
- Görev süreleri deterministiktir.
- Görevler arasındaki öncelik ilişkileri bilinmektedir.
- Bir görevin tamamı bir istasyonda gerçekleştirilmek zorundadır.
- Bir görevin öncülü varsa, o görevin başlayabilmesi için bütün öncüllerinin tamamlanmış olması gerekir.

Problemin temel kısıtları şunlardır:

Tek modelli deterministik görev zamanlı düz MHD probleminin, atama kısıtları, çevrim zamanı kısıtları ve öncelik ilişkileri kısıtları olmak üzere üç çeşit kısıtı vardır.

- *Atama kısıtları:* Montaj hattındaki bütün görevler istasyonlara atanmak zorundadır ve bir görev yalnızca bir istasyona atanabilir.
- *Çevrim zamanı kısıtları:* Montaj hattı üzerinde açılan her bir istasyona atanan görevlerin görev zamanlarının toplamı çevrim zamanını aşamaz.
- *Öncelik ilişkileri kısıtları:* Bir görevin bir istasyona atanabilmesi için o görevin bütün öncüllerinin ya daha önceki bir istasyona ya da üzerinde işlem yapılan istasyona atanmış olması gerekir.

Basit düz MHD probleminin Talbot ve Patterson (1984) tarafından önerilen tamsayılı programlama modeli aşağıda verilmiştir:

$$Minimize \sum_{j=1}^{m_{\max}} jk_{nj} \tag{1}$$

$$\sum_{j=1}^{m_{\max}} k_{ij} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i=1}^{n} t_i k_{ij} \le C \qquad \forall j \tag{3}$$

$$\sum_{j=1}^{m_{\max}} j(k_{rj} - k_{sj}) \le 0 \qquad \forall (r,s) \in S \tag{4}$$

$$k_{ij} \in \{0,1\} \qquad \forall i; \forall j \tag{5}$$

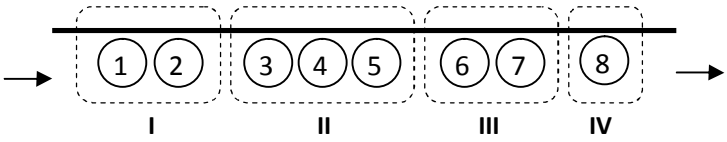
Burada::

i, r, s	: görev
j	: istasyon
C	: çevrim zamanı
m_{max}	: maksimum istasyon sayısı
n	: toplam görev sayısı
t_i	: i görevinin tamamlanma süresi
S	: öncelik ilişkileri kümesi
$(r,s) \in S$	: bir öncelik ilişkisi; r görevi s görevinin komşu öncülüdür
k_{ij}	: 1, i görevi j istasyonuna atanmış ise; 0, aksi halde

Problemin amaç fonksiyonu, öncelik diyagramındaki son görevi mümkün olan en erken istasyona atamak, dolayısıyla montaj hattı boyunca açılan istasyon sayısını minimize etmektir. 2 numaralı kısıt, bütün görevlerin istasyonlara atanmasını ve her görevin bir kere atanmasını sağlamaktadır. 3 numaralı kısıt, açılan bir istasyondaki

görevlerin görev zamanları toplamının çevrim zamanını aşmaması içindir. 4 numaralı kısıt öncelik ilişkileri kısıtı olup, öncülü olan bir görevin ya öncülüyle aynı istasyona ya da öncülünden sonraki bir istasyona atanmasını sağlamaktadır. 5 numaralı kısıt ise modeldeki bütün k_{ij} değişkenlerinin ikili düzende (0–1) olduğunu ifade etmektedir.

Şekil 3.2'de dengelenmiş ve dört iş istasyonundan meydana gelen bir düz montaj hattı görülmektedir.



Şekil 3. 2 Dengelenmiş bir düz montaj hattı

3.1.1.3. U-Tipi montaj hattı dengeleme problemi

U-Tipi MHD problemleri, düz MHD problemlerinin üzerine inşa edilmiştir. Literatürde U-Tipi MHD problemleri ile ilgili mevcut olan çalışmaların hemen hepsinde düz MHD problemleri için önerilen yaklaşımların U-Tipi montaj hatlarına uyarlandığı görülmektedir.

Düz MHD problemleri için kullanılan *atama* ve *çevrim zamanı* kısıtları, U-Tipi MHD probleminde de aynen geçerlidir. Yalnızca *öncelik ilişkileri* kısıtı, U-Tipi montaj hatlarının özel yapısından dolayı farklılık göstermektedir. U-Tipi montaj hatlarında hattın girişiyle çıkışının aynı yönde olması sebebiyle hat içerisinde çalışan bir işçi, hattın hem başlangıç hem de bitiş tarafındaki görevleri yerine getirebilecektir. Buna bağlı olarak, U-Tipi MHD problemi için *öncelik ilişkileri kısıtı* aşağıdaki gibi ifade edilebilir (Miltenburg ve Wijingaard 1994):

- *U-Tipi montaj hatları için öncelik ilişkileri kısıtı:* Bir görevin bir istasyona atanabilmesi için; o görevin bütün öncüllerinin veya bütün ardıllarının daha önce oluşturulan istasyonlara veya üzerinde atama yapılmakta olan istasyona atanmış olması gerekir.

U-Tipi MHD problemlerini düz MHD problemlerinden ayıran en önemli özellik yukarıda belirtilen öncelik ilişkileri kısıtıdır. Düz MHDde, bir istasyona yalnızca, öncülü olmayan veya bütün öncülleri daha önceki istasyonlara atanmış görevler atanabilecekken; U-Tipi MHD'de, bir istasyona hem öncülü olmayan görevler hem de ardılı olmayan görevler atanabilecektir. Dolayısıyla her hangi bir istasyon için, o istasyona atanabilecek görevlerin oluşturacağı küme U-Tipi montaj hatlarında, düz montaj hatlarına göre daha geniş olacaktır. Bunun sonucu olarak bir montaj hattını U-Tipi olarak dengelemek, aynı hattı düz olarak dengelemeye göre daha fazla dengeleme esnekliği sağlamakta ve açılacak istasyon sayısı daha az olabilmektedir.

Aşağıda tek modelli deterministik görev zamanlı U-Tipi MHD probleminin varsayımları, kısıtları ve Urban (1998) tarafından önerilen tamsayılı programlama modeli verilmiştir:

Problemin varsayımları:

Tek modelli deterministik görev zamanlı düz MHD probleminin *öncelik ilişkileri* dışındaki varsayımları tek modelli deterministik görev zamanlı U-Tipi MHD problemi için de aynen geçerlidir. *Öncelik ilişkileri* ile ilgili varsayım ise şu şekilde değişiklik göstermektedir:

- Bir görevin başlayabilmesi için bütün öncüllerinin veya ardıllarının tamamlanmış olması gerekir.

Problemin kısıtları:

Tek modelli deterministik görev zamanlı U-Tipi MHD probleminde, tek modelli deterministik görev zamanlı düz MHD probleminin *atama* ve *çevrim zamanı* kısıtları aynen geçerliliğini korumaktadır. Yalnızca *öncelik ilişkileri* kısıtı farklılık göstermekte olup yukarıda belirtilmiştir.

U-Tipi MHD için Urban (1998) tarafından geliştirilmiş olan tamsayılı programlama modeli aşağıda verilmiştir:

$$\textit{Minimize} \sum_{j=[m_{\min}]+1}^{m_{\max}} z_j \tag{6}$$

$$\sum_{j=1}^{m_{\max}} (x_{ij} + y_{ij}) = 1 \qquad \forall i \tag{7}$$

$$\sum_{i=1}^{n} t_i (x_{ij} + y_{ij}) \le C \qquad j = 1,2,...,[m_{\min}] \tag{8}$$

$$\sum_{i=1}^{n} t_i (x_{ij} + y_{ij}) \le C z_j \qquad j = [m_{\min}]+1,...,[m_{\max}] \tag{9}$$

$$\sum_{j=1}^{m_{\max}} (m_{\max} - j + 1)(x_{rj} - x_{sj}) \ge 0 \qquad \forall (r,s) \in S \tag{10}$$

$$\sum_{j=1}^{m_{\max}} (m_{\max} - j + 1)(y_{sj} - y_{rj}) \ge 0 \qquad \forall (r,s) \in S \tag{11}$$

$$x_{ij}, y_{ij}, z_j \in \{0,1\} \qquad \forall i; \forall j \tag{12}$$

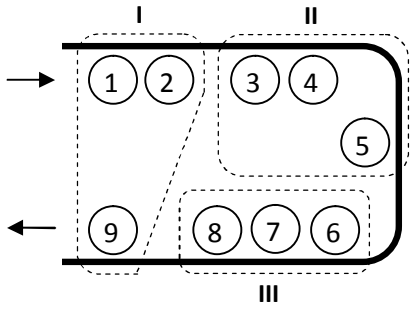
Burada:

i, r, s	: görev
j	: istasyon
C	: çevrim zamanı
m_{max}	: maksimum istasyon sayısı
m_{min}	: minimum istasyon sayısı
n	: toplam görev sayısı
t_i	: *i* görevinin tamamlanma süresi
S	: öncelik ilişkileri kümesi
$(r,s) \in S$	: bir öncelik ilişkisi; *r* görevi *s* görevinin komşu öncülüdür
x_{ij}	: 1, *i* görevi *j* istasyonuna orijinal diyagramdan atanmış ise; 0, aksi halde
y_{ij}	: 1, *i* görevi *j* istasyonuna gölge diyagramdan atanmış ise; 0, aksi halde
z_j	: 1, *j* istasyonu açılmış ise; 0, aksi halde

Yukarıdaki modelin amacı U-Tipi montaj hattı boyunca açılacak istasyon sayısını minimize etmektir. 7 numaralı kısıt *atama* kısıtı olup, bütün görevlerin istasyonlara atanmasını aynı zamanda her görevin orijinal veya gölge diyagramdan olmak üzere yalnızca bir kere atanmasını sağlamaktadır. 8 ve 9 numaralı kısıtlar her

hangi bir istasyona atanan görevlerin görev zamanları toplamının çevrim zamanını aşmayacağını garanti etmektedir. 10 ve 11 numaralı kısıtlar öncelik ilişkileri kısıtlarıdır. 10 numaralı kısıt orijinal diyagramdaki bir görevin ya öncülüyle aynı istasyona ya da öncülünden sonraki bir istasyona atanmasını sağlarken 11 numaralı kısıt, gölge diyagramdaki bir görevin ya ardılıyla aynı istasyona ya da ardılından sonraki bir istasyona atanmasını sağlamaktadır. 12 numaralı kısıt ise modeldeki bütün x_{ij}, y_{ij} ve z_j değişkenlerinin ikili düzende (0–1) olduklarını ifade etmektedir.

Şekil 3.3'de U-Tipi dengelenmiş bir montaj hattı görülmektedir. U-Tipi montaj hatlarında, düz montaj hatlarında olduğu gibi birbiri ardına dizilmiş görevlerden oluşan klasik istasyonların yanı sıra hattın hem ön hem de arka tarafından görevler atanan *karşıya geçişli* istasyonlar da ortaya çıkmaktadır. Şekil 3.3'de görülen I numaralı istasyon karşıya geçişli bir istasyondur.



Şekil 3. 3 Dengelenmiş bir U-Tipi montaj hattı

Modelin yapısı gereği hem öncülü olmayan hem de ardılı olmayan görevlerin istasyonlara atanabilmesi dolayısıyla U-Tipi MHD problemlerinde elde edilebilecek mümkün çözüm sayısı düz MHD problemlerine göre daha fazladır. Bunun dışında da U-Tipi montaj hatlarının düz montaj hatlarına göre çok sayıda avantajı olmakla birlikte bu avantajlar şöyle özetlenebilir (Miltenburg ve Wijingaard 1994):

- İşçiler birbirlerine yakın bir şekilde çalıştıkları için aralarındaki iletişim ve etkileşim daha fazla olmaktadır. Ortaya çıkan her türlü problemde işçiler kolaylıkla birbirlerine yardım edebilirler.
- İşçiler hattın çeşitli istasyonlarında çalıştıkları için kısa sürede çok fonksiyonlu işçi haline gelmektedirler. Böylece talepte meydana gelen

değişmelere kolayca uyum sağlanabildiği gibi işçiler de ortaya çıkabilecek farklı problemleri çözme yeteneği kazanmaktadırlar. Çok fonksiyonlu işçiler farklı süreçler ve operasyonlar arası ilişkileri daha iyi kavrayarak iyileştirme çalışmalarına da daha fazla katkı yapabilmektedirler.

- Talepte meydana gelen değişimlere bağlı olarak U-Tipi montaj hattının tamamlaması gereken ürün miktarı, çok fonksiyonlu işçilerin hatta eklenip çıkarılmasıyla kolayca ayarlanabilir. Geleneksel düz montaj hatlarında hattın çıktısı çalışma saatleri ile ayarlanır.
- Aynı miktarda çıktı elde etmek için U-Tipi montaj hatlarında ihtiyaç duyulan istasyon sayısı düz montaj hatlarında ihtiyaç duyulan istasyon sayısına eşit veya bundan daha az olmaktadır. Hiçbir zaman fazla olamaz.

U-Tipi montaj hatları, yukarıda belirtilen üstünlüklerinin yanı sıra düşük stok seviyeleri, basit malzeme taşıma, basit üretim planlama ve kontrol, yüksek kalite, ekip çalışması gibi çok sayıda avantaj da sağlamaktadır (Kara 2004).

3.1.2. Araştırmada kullanılan test problemleri

Araştırma kapsamında, literatürdeki çeşitli çalışmalardan elde edilmiş, görev sayıları 7 ile 111 arasında değişen 12 adet farklı MHD problemine, farklı çevrim zamanları uygulanarak oluşturulmuş 64 adet test probleminden oluşan bir veri seti kullanılmış ve tüm problemler hem düz hem de U-Tipi MHD problemi olarak çözülmüştür. İlgili veri seti literatürde, Talbot ve ark. (1986) problem seti olarak bilinmektedir ve değişik araştırmalarda kullanılmıştır. Veri setindeki tüm problemlerin öncelik ilişkilerine, görev sürelerine ve ilgili çevrim zamanı için optimal istasyon sayılarına ilişkin verilere http://www.assembly-line-balancing.de adresinden erişilmiştir.

Ayrıca, araştırma kapsamında önerilen BBAOA, Visual Basic 6.0 programlama dilinde kodlanmış ve çözümler Intel Core 2 Duo E7600 3,06 GHz işlemci, 3 Gb RAM'e sahip bir bilgisayarda gerçekleştirilmiştir.

3.2. Metot

3.2.1. Bakteriyel besin arama optimizasyonu algoritması

3.2.1.1. Bakteriyel besin arama optimizasyonu algoritmasının biyolojik temelleri

Escherichia Coli (E. Coli) bakterisi, dünyadaki birçok canlının bağırsaklarında yaşayan bir bakteridir. E. Coli bakterilerinin besin arama sürecindeki davranışlarına yön veren bir kontrol mekanizması bulunmaktadır. Besin arama süreci, besin kaynaklarına doğru gerçekleştirilen bir dizi hareketten oluşmaktadır. Kontrol mekanizması, E. Coli bakterisinin bir sonraki konum değiştirme hareketine referans oluşturacak şekilde bir konumdan diğer konumlara ortaya çıkan değişikliklerin değerlendirilmesi prensibine dayanmaktadır. İki konum arasındaki yer değiştirme bir "hareket" olarak adlandırılır ve her bir hareketin belirli bir yönü ve adım uzunluğu vardır. E. Coli bakterileri, kontrol mekanizmalarına bağlı olarak besin kaynaklarına adım adım yaklaşırlar. E. Coli bakterileri üzerinde yapılan biyolojik çalışmalar, besin arama süreçlerinin dört adımdan oluştuğunu göstermektedir. Bu adımlar şunlardır (Passino 2002):

(1) Makul bir besin bölgesi aramak,

(2) Bulunan makul besin bölgesine girip girmemeye karar vermek,

(3) Yeni bir besin bölgesine girilmişse titiz bir besin araması yapmak ve

(4) Şu an içinde bulunulan besin bölgesindeki besinlerin bir kısmı tüketildikten sonra, yeni ve daha iyi bir besin bölgesine göç etmek veya aynı besin bölgesinde kalmak arasında karar vermektir.

Bakteriler yetersiz besin bulunan bir bölgede kalmışlarsa, önceki tecrübelerine bağlı olarak, genellikle diğer bölgelerin besince zengin olduğu sonucuna varırlar. Bu sonuca bağlı olarak bölgelerini değiştirme eğilimindedirler. Bununla birlikte, her bir konum değiştirme kararı, asıl amaç olan birim zamanda alınan enerjiyi maksimize

etmek doğrultusunda, psikolojik ve çevresel kısıtlar altında verilir. E. Coli bakterilerinin yukarıda bahsedilen besin arama prensiplerini taklit etmek amacıyla Passino (2002) dağıtık optimizasyon ve kontrol için BBAOA'yı geliştirmiştir. Birçok sürü zekâsına bağlı rastsal arama algoritmasında olduğu gibi, BBAOA'da da ilk adım, problemi algoritmaya uygun şekilde kodlamaktır. Sonrasında sürü bilgilerine göre bireyleri ayarlamak ve son olarak optimal çözümü aramaktır. BBAOA'nın bir optimizasyon döngüsü üç olaydan meydana gelir. Bunlar kemotaktik olay, üreme olayı ve eliminasyon-dağılma olayıdır. Bu üç olay aşağıda açıklanmıştır (Passino 2002):

Kemotaktik olay: Mikrobiyolojik çalışmalar, E. Coli bakterilerinin kamçılarıyla hareket ettiklerini göstermektedir. Bütün kamçılar saat yönünün tersine dönerse bakteri ileriye doğru hareket ederken, kamçılar saat yönünde döndüğünde bakteri yavaşlamakta ve bulunduğu yerde salınmaktadır. E. Coli bakterisinin besin araması son iki davranış arasındaki değişimlere bağlıdır.

Bakterinin besin arama sürecinde, kamçıların dönmesi bireyin şu anki ortamının değerlendirilmesine göre olur ve sonrasında şu anki pozisyonun değiştirilip değiştirilmeyeceğine ve bazı parametreler ışığında (bir sonraki hareketin yönü ve adım uzunluğu) nasıl değiştirileceğine karar verilir. BBAOA'da yön değiştirmenin formülü şu şekildedir:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l)+C(i)\varphi(j) \tag{13}$$

Burada $\theta^i(j,k,l)$ *i.* bireyin, şu anki pozisyonunu göstermektedir. *j, k* ve *l,* kemotaktik, üreme ve eliminasyon olaylarının indislerini göstermektedir. $\varphi(j)$ kamçı hareketine bağlı olan hareket yönünü ifade ederken, $C(i)$ adım uzunluğuna karşılık gelmektedir. Ayrıca E. coli bakterisi besin maddesine ulaştığında diğer bakterileri uyarıcı etkiye sahip kimyasal bir madde salgılar. Bu madde, diğer E. coli bakterilerinin besini bulan bakterinin bulunduğu yere doğru hareket etmesini sağlar. Bu olaya sürü zekası veya sürü davranışı adı verilmektedir.

Üreme olayı: Bir besin arama periyodundan sonra, bazı bakterilerin besin arama stratejilerinin açıkça başarısız olduğu ortaya çıkmaktadır. Bu bakteriler, yeterli besin bulma ihtimallerinin az olması nedeniyle popülasyondan çıkarılmaktadır. Popülasyon sayısını sabit tutmak için, çıkarılanların yerini almak üzere arama stratejisi iyi olan bakteriler, aynı miktarda kopyalanmaktadır. Üreme olayı şu şekilde uygulanır:

Popülasyondaki bakteri sayısı S ise, $S/2$ sayıda bakteri popülasyondan çıkarılır. Öncelikle her bir bakteri, konumlarındaki değerlendirme kriterine göre sıralanır. Sonrasında, sıralamanın son yarısındaki bakteriler popülasyondan çıkarılarak ilk yarısındakilerin her birinin aynı konumda yer alacak şekilde bir kopyası alınır. Bu işlem bir nevi bakterilerin bölünerek çoğalmalarının taklit edilmesidir.

Eliminasyon-dağılma olayı: Bakterilerin yaşadığı ortamdaki, aşırı derecede ısı artışı, hızlı su akıntıları ve bunun gibi diğer etkenler, bakterilerin davranışlarını büyük ölçüde etkilemektedir. Bütün bu etkenler, popülasyonda ani veya yavaş değişikliklere neden olabilir. Bu değişiklikler, o bölgedeki bütün bakterilerin ölümü veya bir kısmının başka bir bölgeye sürüklenmesi veya göç etmesi şeklinde olabilir. Eliminasyon-dağılma olayı, bu biyolojik süreçleri taklit etmek amacıyla uygulanmaktadır. Bu uygulama, kemotaktik olayın performansına olumsuz etki edebileceği gibi, bireyleri daha iyi bir besin bölgesinin yakınına sürükleyerek olumlu da etki edebilmektedir. Eliminasyon-dağılma olayı bir P_{ed} olasılığına bağlı olarak gerçekleştirilir. Belirli bir birey dağılma olayına maruz kalırsa, o birey yok edilir ve yeni bir birey üretilir. Bu uygulama, o bireyin yeni bir konuma hareket etmesi manasına gelmektedir.

BBAOA'nın temel adımları aşağıda verilmiştir (Passino 2002):

Adım 1: Popülasyonu oluştur

Adım 2: Bireyleri değerlendirme fonksiyonuna göre değerlendir

Adım 3: Optimizasyon için üç döngü:

İç döngü: Kemotaktik olay

Orta döngü: Üreme olayı

Dış döngü: Eliminasyon-dağılma olayı

Adım 4: Son çözümü belirlemek için optimal bireyin kodunu çöz

İç döngü olan kemotaktik olay, üç döngünün çekirdeği niteliğindedir. Bir canlı için besin arama, yeni bir bölgeye girip girmemeye karar verme, şu anki bölgede kalma süresi, bir sonraki adımda seçilecek yön gibi kararların verilmesini sağlamaktadır. Bu nedenle, kemotaktik olay algoritmanın çözüme yaklaşması konusunda çok önemli bir etkiye sahiptir.

Algoritmanın başlangıç değerlerini belirlemek için, p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed} ve $C(i)$, $i = 1,2,\ldots..,S$ değişkenlerinin değerlerinin atanması gerekir. Eğer sürü davranışı adımı da kullanılacaksa $d_{attract}$, $w_{attract}$, $h_{repellant}$, $w_{repellant}$ hücreden hücreye çekim fonksiyonu değerleri de belirlenmelidir. Aynı zamanda θ^i için başlangıç değerleri seçilmelidir, $i = 1,2,\ldots..,S$. Bu değerler optimum değerin olduğu alanlardan seçilebileceği gibi optimizasyon alanına çözüm uzayı boyunca rasgele dağıtılabilir. Bakteri popülasyonunun kemotaksis, üreme, eliminasyon ve dağılma olaylarının modellendiği algoritma aşağıdaki gibidir (başlangıç olarak, $j = k = l = 0$ olarak alınır) (Passino 2002):

1) Eliminasyon ve dağılma döngüsü: $l = l + 1$
2) Üreme döngüsü: $k = k + 1$
3) Kemotaksis döngüsü: $j = j + 1$
   a) $i = 1,2,....,S$, için i. Bakteri için bir kemotaktik adım başlat.

   b) $J(i,j,k,l)$'yi hesapla.

$$J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l)) \tag{14}$$

   hücreden hücreye çekici etkisini besin konsantrasyon değerine ekle.

   c) $J_{last} = J(i,j,k,l)$ atamasını gerçekleştir ve yüzme işlemi aracılığı ile daha iyi

   bir sonuç elde edinceye kadar bu değeri sakla.

d) *Yuvarlanma*: $\Delta_m(i)$, m = 1, 2, …., p olmak üzere [-1, 1] aralığında rasgele bir $\Delta(i) \in \Re^p$ vektörü oluştur.

e) *Hareket etme*:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{15}$$

atamasını gerçekleştir. Bu sonuç, *i*. Bakteri için $C(i)$ adım büyüklüğü kadar ve yuvarlanma yönünde olacaktır.

f) $J(i, j+1, k, l)$'yi hesapla.

$$J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l)) \tag{16}$$

g) *Yüzme*:

i) Yüzme uzunluğunun sayacı olmak üzere m = 0 ilk değerini ata.

ii) $m < N_s$ olduğu sürece;

- $m = m + 1$ atamasını gerçekleştir.
- Eğer $J(i, j+1, k, l) < J_{last}$ ise, yani daha iyi bir sonuç ise, $J_{last} = J(i, j+1, k, l)$ ve

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{17}$$

atamalarını gerçekleştir. Yeni $J(i, j+1, k, l)$ değerini hesaplamak için bu $\theta^i(j+1,k,l)$ değerini kullan.

- $m = N_s$ ise bu bakteri için yüzme sürecini durdur.

h) Son bakteri değilse ($i \neq S$) bir sonraki bakteriyi seç ($i = i + 1$) ve "b" maddesine git.

4) Eğer $j < N_c$ ise bakterilerin yaşam süresi bitmemiştir. 3. adıma tekrar dön ve kemotaktik adımlara devam et.

5) Üreme İşlemi:

a) Hali hazırdaki k ve l değerleri ve her bir i = 1, 2, …, S, değeri için,

$$J_{health} = \sum_{j=1}^{N_c+1} J(i, j, k, l) \tag{18}$$

bir bakterinin yaşadığı sürece ne kadar besin aldığını ve zararlı maddelerden uzak durmada ne kadar başarılı olduğunu ifade eden fonksiyondur. Bir başka deyişle i bakterisinin sağlık durumunu ifade etmektedir. Sonuçta elde edilen J_{health} değerlerine göre bakterileri ve kemotaktik parametrelerini küçükten büyüğe doğru sırala.

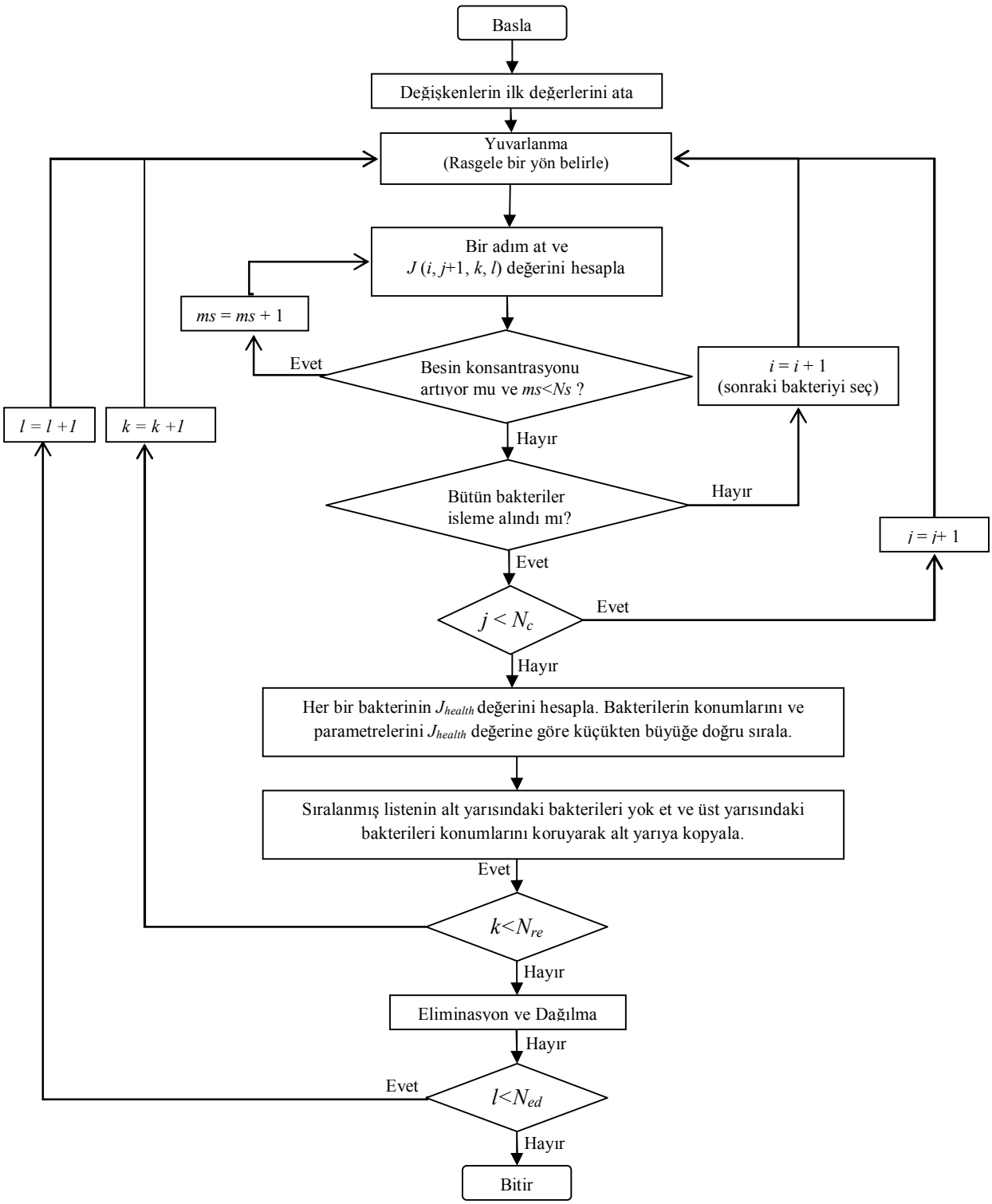
b) Sıralamanın alt yarısındaki sağlık durumları diğerlerine göre daha kötü olan $S_r = S/2$ adet bakteri ölmüş olarak kabul edilir. Bu durumda sağlık durumu daha iyi olan ve sıralamanın üst kısmında kalan Sr adet bakteri bölünerek üretilir. Ebeveynleri ile aynı konumlara yerleştirilecek şekilde, yeni nesil bakteri çiftlerinden her bir tanesi listenin alt kısmındaki ölen bakterilerin yerine geçirilir.

6) Eğer $k < N_{re}$ ise belirlenen üreme üst sınırına ulaşılmamış demektir, 2. adıma geri dön ve sonraki nesil ile kemotaktik adımları yerine getir.

7) Eliminasyon ve Dağılma: i = 1, 2, ….., S, dizisi içerisindeki bütün bakterileri P_{ed} olasılığına bağlı kalarak eliminasyon ve dağılma işlemine tabi tut. Bu işlem, popülasyon adedini sabit tutmak şartıyla bir bakterinin basit bir şekilde bulunduğu yerden rasgele yeni bir yere atılması ile gerçekleştirilir.

8) Eğer $l < N_{ed}$ ise 1.adıma git, aksi takdirde programı sonlandır (Passino 2002, Başbuğ 2008).

BBAOA'nın akış diyagramı Şekil 3.4'te verilmiştir (Başbuğ 2008).



Şekil 3. 4 BBAOA'nın akış diyagramı (Başbuğ 2008)

3.2.1.2. Çözüm kodlama, görev atama prosedürü ve başlangıç çözümleri

Önerilen BBAOA'da her bir bakteri, ilgili problemin görevlerinden ve bu görevlerin atandığı istasyon bilgilerinden oluşan bir çözümü temsil etmektedir. Hem düz hem de U-Tipi MHD problemlerinin çözümünde, ilgili problemin görev sayısı kadar elemana sahip bir görev sırası oluşturulmakta ve sıradaki her bir pozisyon görev numarası ile ifade edilmektedir. Buna bağlı olarak, sıradaki her bir pozisyona karşılık, ilgili pozisyondaki görevin atandığı istasyon bilgisi de hafızada tutulmaktadır. Şekil 3.5a'da, 7 görevli bir problem için kodlanmış bir çözüm verilmiştir ve her bir pozisyondaki sayı görev numarasını ifade etmektedir. Şekil 3.5b'de ise, Şekil 3.5a'da gösterilen kodun çözümü verilmiştir ve her bir pozisyon için ilk kısım görev numarasını, ikinci kısım ise bu görevin atandığı istasyonu göstermektedir.

1	4	2	5	6	3	7

a) Kodlanmış bir çözüm

1–1	4–1	2–1	5–2	6–3	3–3	7–4

b) Kodun çözümü

Şekil 3. 5 Örnek bir çözüm kodlaması ve ilgili kodun çözümü

Görev sırasının herhangi bir pozisyonuna yeni bir görev eklenirken daha önceki pozisyonlara eklenmiş görevler istasyonlara atanarak ilerlenmektedir. Bir görev sıraya ekleneceğinde, daha önce eklenmiş görevlere bağlı olarak öncelik diyagramından geçici bir atanabilir görevler kümesi oluşturulmaktadır. Bu kümedeki görevler içerisinden, görev süresi itibariyle üzerinde işlem yapılan istasyona atanabilecek olan görevler belirlenmekte ve elde edilen görevler arasından bir görev seçilerek görev sırasının ilgili pozisyonuna eklenmektedir. Eğer öncelik ilişkilerine bağlı olarak belirlenen geçici atanabilir görevler kümesindeki hiçbir görev, görev süresi itibariyle üzerinde işlem yapılan istasyona atanamıyorsa yeni bir istasyon açılır. Algoritmanın ilgili prensiplerine bağlı olarak işleyen bu süreç sonunda,

problemin bütün görevlerinden ve bu görevlerin atandığı istasyon bilgilerinden oluşan bir görev sırası elde edilmektedir.

Her görevin, önceden belirlenen maksimum istasyon sayısı kadar istasyonun her birine atanmasında öncelik kuralı olarak kullanılmak üzere bir ağırlık değeri bulunmaktadır. Görevler istasyonlara bu ağırlık değerleri dikkate alınarak atanmaktadır. Başlangıç çözümlerinde tüm ağırlık değerleri çevrim zamanına eşit olarak kabul edilmiştir. Bu durumda, başlangıç çözümleri elde edilirken, üzerinde işlem yapılan herhangi bir istasyona atanabilir görevler kümesinden rastsal olarak seçim yapılmaktadır.

Maksimum istasyon sayısı, teorik minimum istasyon sayısının 1,3 ile çarpılarak sonucun bir üst tam sayıya yuvarlanması ile elde edilmiştir. Teorik minimum istasyon sayısı ise MHD literatüründe sıklıkla kullanılan bir kavram olup, ilgili problemin görev süreleri toplamının çevrim zamanına bölünmesiyle elde edilen değerin bir üst tam sayıya yuvarlanması ile hesaplanmaktadır.

Bir problem için elde edilen bir çözümün maliyeti Eşitlik 19'de verilen formülle hesaplanmaktadır.

$$J = (n-1)C + t_{son} \tag{19}$$

Bu ifadedeki J; çözümün maliyetini (amaç fonksiyonu değeri), n; istasyon sayısını, C; çevrim zamanını t_{son} ise son istasyonun iş yükünü temsil etmektedir. Miltenburg (1998) tarafından da kullanılan ve amaç fonksiyonu olarak minimize edilmeye çalışılan bu formülasyon sayesinde, son istasyondan önceki istasyonlar mümkün olduğunca doldurulmakta, son istasyonun iş yükü de maliyete dâhil edilmek suretiyle çözüm süreci ilerledikçe son istasyon boşaltılmakta ve toplam istasyon sayısı azaltılmaktadır.

3.2.1.3. Kemotaksis

Orijinal BBAOA'da olduğu gibi her bir kemotaksis başlangıcında, görev sayısı uzunluğunda ve [-1,1] aralığında, önceden belirlenen maksimum istasyon sayısı kadar rastsal vektör geliştirilmektedir. Bu geliştirilen vektörlere bağlı olarak belirlenen bakteri yönü, adım uzunluğuyla çarpıldıktan sonra ilgili bakterinin temsil ettiği çözümdeki her bir istasyon için ayrı ayrı uygulanarak, ilgili görevlerin istasyonlara atanma işleminde kullanılacak ağırlıklar güncellenmektedir.

Güncellenmiş ağırlıklara bağlı olarak atama işlemi gerçekleştirilirken, atanabilir görevler kümesi içerisindeki her bir görevin, üzerinde işlem yapılan istasyona karşılık gelen ağırlık değerlerine bakılarak, en yüksek ağırlık değerine sahip görev ilgili istasyona atanmaktadır. Bütün görevlerin atanmasından sonra elde edilen amaç fonksiyonu değeri, ilgili bakterinin bir önceki amaç fonksiyonu değerinden küçükse elde edilen bu yeni çözüm, ilgili bakterinin mevcut çözümü olarak kabul edilmekte ve aynı yön vektörleri bu bakteriye tekrar uygulanarak aynı yönde bir adım daha ilerlemesi sağlanmaktadır. Bu durum, ilgili bakterinin çözümü iyileşmeyinceye veya adım sayısı sınırına ulaşılıncaya kadar tekrarlanmaktadır.

3.2.1.4. Üreme

Önerilen BBAOA'da kullanılan üreme olayı, orijinal BBAOA'da olduğu gibi, bakterilerin yaşam süreleri boyunca elde ettikleri amaç fonksiyonu değerleri toplamlarına göre sıralanarak, kötü yarısının popülasyondan atılıp, iyi yarısının kopyalanmasıyla gerçekleştirilmektedir.

3.2.1.5. Eliminasyon ve dağılma

Önerilen BBAOA'nın eliminasyon-dağılma olayında, her bir bakteri için (0,1) aralığında rastsal bir sayı geliştirilmekte ve bu sayı daha önceden belirlenen eliminasyon-dağılma olasılığından küçükse, ilgili bakteri için, başlangıç çözümünde olduğu gibi çevrim zamanına eşit yeni ağırlıklar geliştirilmekte atanabilir görevler kümesinden rastsal seçim yapılarak yeni bir çözüm elde edilmektedir.

3.2.1.6. Açıklayıcı bir örnek

Bu bölümde, Şekil 3.1'de öncelik diyagramı verilen Merten (1967) probleminin C=10 çevrim zamanı için bir bakteriye ait başlangıç çözümü ve bir kemotaksis süreci örnek üzerinde açıklanmıştır. Tablo 3.1'de ilgili problemin görev süreleri verilmiştir.

Tablo 3. 1 Merten (1967) problemi için görev süreleri

Görev No	Görev Süresi
1	1
2	5
3	4
4	3
5	5
6	6
7	5

İlgili problem için teorik minimum istasyon sayısı 3, maksimum istasyon sayısı ise 4 olarak belirlenmiştir. Bu durumda, her bir görevin her bir istasyona atanması işleminde kullanılmak üzere başlangıçta çevrim zamanına eşit olarak belirlenen ağırlıklar ise Tablo 3.2'de verilmiştir.

Tablo 3. 2 Atama ağırlıkları için örnek başlangıç değerleri

İstasyon \ Görev	1	2	3	4	5	6	7
1	10	10	10	10	10	10	10
2	10	10	10	10	10	10	10
3	10	10	10	10	10	10	10
4	10	10	10	10	10	10	10

İlgili problem için bir bakteriye ait başlangıç çözümü şu şekilde oluşturulur:

- *İstasyon* = 1; İstasyondaki boş zaman BZ = 10; Öncelik diyagramına bağlı geçici atanabilir görevler kümesi, $GAGK_1$ ={1}; Boş zamana bağlı olarak atanabilir görevler kümesi, AGK_1 = {1}; AGK içersinden rastsal seçilen görev, ***SG* = 1**;BZ=BZ–süre(SG); BZ = 10–1 = 9
- $GAGK_2$ = {2,4}; AGK_2 = {2,4}; ***SG* = 4;** BZ = 9 – 3 = 6
- $GAGK_3$={2,7}; AGK_3={2,7}; ***SG*=2;** BZ = 6 – 5 = 1
- $GAGK_4$={3,5,7}; AGK_4 = Ø ⇨ *İstasyon* = *İstasyon* + 1 = 2; BZ=10
- $GAGK_4${3,5,7}; AGK_4={3,5,7}; ***SG*=5**; BZ = 10 – 5 = 5
- $GAGK_5$={3,6,7}; AGK_5={3,7}; ***SG*=3**; BZ = 5 – 4 = 1
- $GAGK_6$={6,7}; AGK_6 = Ø ⇨ *İstasyon* = *İstasyon* + 1 = 3; BZ=10
- $GAGK_6$={6,7}; AGK_6={6,7}; ***SG*=6**; BZ = 10 – 6 = 4
- $GAGK_7$={7}; AGK_7 = Ø ⇨ İstasyon = *İstasyon* + 1 = 4; BZ=10
- $GAGK_7$={7}; AGK_7={7}; ***SG*=7**; BZ = 10 – 5 = 5; son istasyonun iş yükü = 5.

Bu durumda, Merten (1967) problemin için, çevrim zamanı 10 kabul edildiğinde, bir bakteriye ait başlangıç çözümünün kodu ve çözümü Şekil 3.6'da verilmiştir.

1	4	2	5	3	6	7

a) Kodlanmış başlangıç çözümü

1–1	4–1	2–1	5–2	3–2	6–3	7–4

b) Kodun çözümü

Şekil 3. 6 Örnek problem için bir bakteriye ait başlangıç çözümü

Elde edilen başlangıç çözümünün amaç fonksiyonu değeri Eşitlik 19'a göre aşağıdaki gibi hesaplanır:

$$J = (4 - 1) \times 10 + 5 = 35$$

Başlangıç çözümü oluşturulduktan ve amaç fonksiyonu değeri hesaplandıktan sonra, parametreler başlangıç değerlerine getirilir ve BBAOA başlar. Algoritmanın başlamasını takip eden kemotaksis sürecinde, öncelikli olarak yön belirleme, ardından ise belirlenen yönde bir adım atma işlemi gerçekleştirilir. Yön belirleme işleminde ise öncelikle her bir istasyon için problemin görev sayısı uzunluğunda ve [-1,1] aralığında rastsal olarak üretilmiş değerlerden oluşan vektörler oluşturulur. Oluşturulan bu vektörler kullanılarak, Eşitlik 20'ye göre bakterinin atama ağırlıklarını güncelleme işleminde kullanılmak üzere, her bir istasyona ait yönler elde edilir.

$$\varphi(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{20}$$

Eşitlik 20'deki $\varphi(i)$ ifadesi i bakterisi için belirlenen yönü, $\Delta(i)$ ifadesi, i bakterisi için [-1,1] aralığında rastsal olarak geliştirilmiş vektörü ifade etmektedir. Önerilen BBAOA'da, rastsal vektör belirleme ve bu vektörü Eşitlik 20 yardımıyla yön vektörüne dönüştürme işlemi her bir istasyon için tekrar edilmektedir.

Tablo 3.3'de örnek problemde bir bakteriye ait her bir istasyon için [-1,1] aralığında rastsal olarak geliştirilmiş vektörler kullanılarak Eşitlik 20 yardımıyla oluşturulan yön vektörleri verilmiştir.

Tablo 3. 3 Örnek problem için elde edilen yön vektörleri

	Görevler						
	1	2	3	4	5	6	7
1. istasyon için yön vektörü:	[0,12	0,26	-0,14	0,44	-0,19	-0,52	0,02]
2. istasyon için yön vektörü:	[0,04	-0,30	-0,41	-0,22	0,82	0,11	0,63]
3. istasyon için yön vektörü:	[0,32	0,13	0,54	-0,42	0,11	-0,56	-0,21]
4. istasyon için yön vektörü:	[0,17	-0,46	-0,81	0,19	0,32	0,05	-0,66]

Tablo 3.3'de gösterilen yön vektörlerindeki her bir değer, adım uzunluğuyla çarpılarak ilgili görevin ilgili istasyona atanmasında dikkate alınacak atama ağırlığına eklenmiş ve güncellenmiş atama ağırlıkları elde edilmiştir. Önerilen BBAOA ile yapılan çözümlerde, adım uzunluğu ilgili problemin çevrim zamanına eşit kabul edildiğinden, örnek problem için adım uzunluğu 10 olarak alınmıştır. Bir örnek vermek gerekirse; Tablo 3.3'de görüldüğü gibi, 1 numaralı görevin 1 numaralı istasyona atanma ağırlığını güncellemede kullanılacak yön değeri 0,12'dir. Bu değer adım uzunluğu olan 10 ile çarpıldığında elde edilen 1,2 değeri, Tablo 3.2'de verilmiş olan, 1 numaralı görevin 1 numaralı istasyona atanma işleminde kullanılacak atama ağırlığına eklenmiştir. Tablo 3.3'teki yön vektörleri kullanılarak elde edilen güncellenmiş atama ağırlıkları Tablo 3.4'te gösterilmiştir.

Tablo 3. 4 Örnek problem için güncellenmiş atama ağırlıkları

		Görev						
		1	2	3	4	5	6	7
İstasyon	1	11,2	12,6	8,6	14,4	8,1	4,8	10,2
	2	10,4	7	5,9	7,8	18,2	11,1	16,3
	3	13,2	11,3	15,4	5,8	11,1	4,4	7,9
	4	11,7	5,4	1,9	11,9	13,2	10,5	3,4

Kemotaksis sürecinde, Tablo 3.4'te verilen güncellenmiş ağırlıklara göre yeni bir çözüm ise şu şekilde oluşturulmaktadır:

- *İstasyon = 1*; İstasyondaki boş zaman BZ = 10; Öncelik diyagramına bağlı geçici atanabilir görevler kümesi, $GAGK_1$ ={1}; Boş zamana bağlı olarak atanabilir görevler kümesi, AGK_1 = {1}; AGK içersinden ilgili istasyona atama ağırlığı en büyük olan ve atanmak üzere seçilen görev, ***SG* = 1**; BZ=BZ–süre(SG); BZ = 10 – 1 = 9

- $GAGK_2$ = {2,4}; AGK_2 = {2,4}; ***SG* = 4;** BZ = 9 – 3 = 6

- $GAGK_3$={2,7}; AGK_3={2,7}; ***SG*=2;** BZ = 6 – 5 = 1
- $GAGK_4$={3,5,7}; AGK_4 = Ø ⇨ *İstasyon* = *İstasyon* + 1 = 2; BZ=10
- $GAGK_4${3,5,7}; AGK_4={3,5,7}; ***SG*=5**; BZ = 10 – 5 = 5
- $GAGK_5$={3,6,7}; AGK_5={3,7}; ***SG*=7**; BZ = 5 – 5 = 0
- $GAGK_6$={3,6}; AGK_6 = Ø ⇨ *İstasyon* = *İstasyon* + 1 = 3; BZ=10
- $GAGK_6$={3,6}; AGK_6={3,6}; ***SG*=3**; BZ = 10 – 4 = 6
- $GAGK_7$={6}; AGK_7={6}; ***SG*=6**; BZ = 6 – 6 = 0; son istasyonun iş yükü = 10.

Örnek problem için kemotaksis sürecinde bir adım atıldıktan sonra elde edilen çözümün kodu ve bu kodun çözümü Şekil 3.7'de verilmiştir.

1	4	2	5	7	3	6

a) Kemotaksis sürecinde elde edilmiş çözümün kodu

1–1	4–1	2–1	5–2	7–2	3–3	6–3

b) Kodun çözümü

Şekil 3. 7 Örnek problem için kemotaksis sürecinde elde edilen çözüm

Şekil 3.7b'de verilmiş olan, örnek problem için bir bakteriye ait kemotaksis sürecinde elde edilmiş çözümün kodu incelendiğinde, toplam istasyon sayısının 3, son istasyonun iş yükünün ise 10 olduğu görülmektedir. İlgili çözümün amaç fonksiyonu değeri ise Eşitlik 19'a göre şu şekildedir:

$$J = (3 - 1) \times 10 + 10 = 30$$

Bu amaç fonksiyonu değerinin, aynı bakterinin başlangıç çözümünde ulaştığı amaç fonksiyonu değeri olan 35'den daha küçük olduğu görülmektedir. Bu durumda yeni elde edilen çözümün kodu bakterinin mevcut çözümü, amaç fonksiyonu değeri

bakterinin mevcut amaç fonksiyonu değeri ve bu çözümü elde etmekte kullanılan atama ağırlıkları bakterinin mevcut atama ağırlıkları olarak kabul edilir. Bakterinin aynı yönde bir adım daha atmasını sağlamak amacıyla aynı vektörleri tekrar kullanılarak atama ağırlıkları tekrar güncellenerek yeni bir çözüm elde edilir. Bu işlem çözüm iyileştikçe ve adım sayısı sınırına ulaşılmadığı sürece devam eder. Yeni elde edilen çözüm, bakterinin mevcut çözümünden daha iyi değilse veya adım sayısı sınırına ulaşılmışsa, bir sonraki bakteri ile işlemlere geçilir ve aynı bakteriye tekrar sıra geldiğinde yeni bir yön belirlemek amacıyla yeni yön vektörleri geliştirilir.

3.2.2. Deney Tasarımı

Önerilen BBAOA'da, bakteri sayısı (S), kemotaktik süreç (kemotaksis) sayısı (N_k), üreme sayısı ($N_{ür}$), eliminasyon-dağılma sayısı (N_{ed}), eliminasyon-dağılma olasılığı (P_{ed}), adım uzunluğu (c(i)) ve adım sayısı sınırı (N_s) olmak üzere yedi farklı parametre bulunmaktadır. Algoritmanın kodlanması, geçerliliğinin test edilmesi ve geliştirilmesi süreçlerinde, parametre seviyelerine ilişkin yapılan denemeler ve tecrübelerle, çözümlere önce bakteri sayısı ve kemotaksis sayısının çok küçük değerleriyle başlanması, üreme ve eliminasyon-dağılma olaylarının ise başlangıçta kullanılmaması uygun görülmüştür. Çözümler, 2 bakteri 2 kemotaksis ile başlamış, problemin bilinen optimal sonucuna ulaşılamaması durumunda, öncelikle kemotaksis sayısı ikişer ikişer artırılmış, kemotaksis 10 değerine ulaştıktan itibaren ise bakteri sayısı ikişer ikişer artırılmıştır. Bakteri ve kemotaksis sayılarının her ikisi de 10 alınmış ve halen optimal çözüme ulaşılamamışsa bir üreme eklenmiştir. Üreme ekledikten sonra da optimal çözüm bulunamamışsa tekrar kemotaksis sayısı ikişer ikişer artırılma yoluna gidilmiştir. Bu durumda da kemotaksis 20 değerine ulaştıktan sonra bakteri sayısı tekrar artırılmaya başlamıştır. Bakteri ve kemotaksis sayılarının her ikisi de 20 olduğunda bir üreme daha eklenmiştir. Optimal sonuca ulaşılamadıkça devam eden bu işlemler, bakteri ve kemotaksis sayılarının her ikisi de ilgili problemin görev sayısından büyük veya eşit oluncaya kadar tekrarlanmış, en son işlem olarak ise 1 eliminasyon-dağılma olayı eklenerek tekrar çözüm yapılmıştır. 20

görevli bir problem için tüm denemelerde optimal sonuca ulaşılamaması durumunda ortaya çıkacak parametre seviyeleri Tablo 3.5'de verilmiştir.

Ayrıca, bütün problemler için, her bir denemede parametre seviyeleri sabit tutularak üç defa çözüm tekrarlanmış, herhangi birinde optimal sonuca ulaşılması durumunda ilgili çözüm ve parametreleri seviyeleri kabul edilmiş, üç denemede de optimale ulaşılamaması durumunda bir sonraki denemeye geçilmiştir.

Tablo 3. 5 20 görevli bir problem için örnek parametre seviyeleri

Deneme No	S	N_k	$N_{ür}$	N_{ed}
1	2	2	0	0
2	2	4	0	0
3	2	6	0	0
4	2	8	0	0
5	2	10	0	0
6	4	10	0	0
7	6	10	0	0
8	8	10	0	0
9	10	10	0	0
10	10	10	1	0
11	10	12	1	0
12	10	14	1	0
13	10	16	1	0
14	10	18	1	0
15	10	20	1	0
16	12	20	1	0
17	14	20	1	0
18	16	20	1	0
19	18	20	1	0
20	20	20	1	0
21	20	20	2	0
22	20	20	2	1

Tüm çözümlerde, adım uzunluğu $c(i)$, ilgili problemin çevrim zamanına eşit kabul edilmiştir. Adım sayısı sınırı N_s ise 10 alınmıştır. Eliminasyon-dağılma kullanılan çözümlerde, eliminasyon-dağılma olasılığı P_{ed}, 0,1 olarak kabul edilmiştir. Önerilen BBAOA'da sürü zekası kullanılmamıştır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde, araştırma kapsamında ele alınan test problemlerinin önerilen BBAOA için düz ve U-Tipi MHD sonuçları, ilgili sonuçlara ulaşıldığı andaki parametre seviyeleri ve çözüm sürelerinin yanı sıra, aynı problemlerin literatürde bilinen optimal sonuçları ve Baykasoğlu ve Dereli (2009) tarafından önerilen KKOA sonuçları ile karşılaştırması verilmiştir. Buna ilave olarak U-Tipi dengeleme göz önünde bulundurulan problemlerde elde edilen sonuçlar, Ajenblit ve Wainwright (1998) tarafından önerilen GA sonuçları ile karşılaştırılmıştır.

Tablo 4. 1 Önerilen BBAOA için düz MHD sonuçları

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Optimal İst. Sayısı	BBAOA Sonucu	Amaç Fonksiyonu	Parametreler				CPU (sn)
					S	N_k	$N_{ür}$	N_{ed}	
Merten (7)	6	6	6	35	2	2	0	0	0,03
	7	5	5	33	2	2	0	0	0,01
	8	5	5	36	2	2	0	0	0,01
	10	3	3	30	2	2	0	0	0,01
	15	2	2	29	2	2	0	0	0,01
	18	2	2	29	2	2	0	0	0,01
Bowman (8)	20	5	5	93	2	2	0	0	0,01
Jaeschke (9)	6	8	8	48	2	2	0	0	0,02
	7	7	7	48	2	2	0	0	0,01
	8	6	6	46	2	2	0	0	0,01
	10	4	4	40	2	2	0	0	0,01
	18	3	3	42	2	2	0	0	0,01
Mansoor (11)	48	4	4	188	2	2	0	0	0,01
	62	3	3	186	2	2	0	0	0,01
	94	2	2	185	2	2	0	0	0,01
Jackson (11)	7	8	8	53	2	2	0	0	0,02
	9	6	6	54	2	2	0	0	0,01
	10	5	5	49	2	2	0	0	0,01
	13	4	4	48	2	2	0	0	0,02
	14	4	4	51	2	2	0	0	0,01
	21	3	3	46	2	2	0	0	0,02
Mitchell (21)	15	8	8	115	2	2	0	0	0,06
	14	8	8	110	4	10	0	0	0,37
	21	5	5	105	2	2	0	0	0,06
	26	5	5	120	2	2	0	0	0,06
	35	3	3	105	2	2	0	0	0,06
	39	3	3	109	2	2	0	0	0,05
Heskia (28)	138	8	8	1090	2	2	0	0	0,14
	205	5	5	1024	20	20	1	0	14,04
	216	5	5	1063	2	2	0	0	0,12
	256	4	4	1024	10	20	1	0	7,06
	324	4	4	1044	2	2	0	0	0,12
	342	3	3	1025	2	4	0	0	0,32

Tablo 4. 1 (Devam)

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Optimal İst. Sayısı	BBAOA Sonucu	Amaç Fonksiyonu	Parametreler				CPU (sn)
					S	N_k	$N_{ür}$	N_{ed}	
Sawyer (30)	25	14	14	348	2	2	0	0	0,15
	27	13	13	347	2	10	0	0	0,50
	30	12	12	353	2	2	0	0	0,14
	36	10	10	340	2	2	0	0	0,16
	41	8	8	328	16	20	1	0	14,47
	54	7	7	340	2	2	0	0	0,17
	75	5	5	334	2	2	0	0	0,12
K. & Wester (45)	57	10	10	567	2	4	0	0	0,93
	79	7	7	552	20	20	1	0	56,06
	92	6	6	552	10	10	1	0	15,67
	110	6	6	555	2	2	0	0	0,59
	138	4	4	552	2	6	0	0	1,03
	184	3	3	552	2	6	0	0	1,03
Tonge (70)	176	21	21	3681	10	10	1	0	58,62
	364	10	10	3596	2	2	0	0	1,49
	410	9	9	3645	2	2	0	0	1,47
	468	8	8	3579	2	2	0	0	1,39
	527	7	7	3562	2	2	0	0	1,88
Arcus (83)	5048	16	16	80711	2	2	0	0	2,24
	5853	14	14	79780	2	2	0	0	2,32
	6842	12	12	78953	2	2	0	0	2,65
	7571	11	11	79401	2	2	0	0	2,29
	8412	10	10	79399	2	2	0	0	2,31
	8898	9	9	78275	2	2	0	0	2,27
	10816	8	8	79403	2	2	0	0	2,63
Arcus (111)	5755	27	28*	160538	2	2	0	0	5,83
	8847	18	18	157777	2	2	0	0	6,32
	10027	16	16	157313	2	2	0	0	5,46
	10743	15	15	155721	2	2	0	0	5,87
	11378	14	14	158677	2	2	0	0	6,56
	17067	9	9	152560	2	2	0	0	7,23

* En yüksek parametre seviyelerine ulaşıncaya kadar deneme yapılmış ancak optimal istasyon sayısına ulaşılamamıştır. Optimalden bir fazla istasyon sayısıyla elde edilen çözümler için en düşük parametre seviyeleri ve ilgili çözüm süreleri gösterilmiştir.

Tablo 4.1'de önerilen BBAOA'nın düz MHD problemi sonuçları ve ilgili çözümlere ulaşıldığı andaki parametre seviyeleri ile çözüm süreleri verilmiştir. Problem adının yanında parantez içindeki sayı, problemin görev sayısını ifade etmektedir. Tablo 4.1 incelendiğinde, önerilen BBAOA'nın ilgili test problemleri üzerinde oldukça etkili olduğu, optimal sonuçlara çok kısa süreler içerisinde ulaştığı görülmektedir. Önerilen BBAOA ile Tablo 4.1'de verilen 64 adet test probleminin 63'ünde optimal sonuçlara saniyelerle ifade edilen sürelerde ulaşılmış, sadece bir problemde, daha önce belirtildiği üzere problemin görev sayısına bağlı olarak ortaya çıkan en yüksek parametre seviyelerine kadar çözümler yapılmasına rağmen optimal sonuçlara ulaşılamamıştır. Önerilen BBAOA ile, düz MHD problemleri için optimal

istasyon sayısına ulaşılamayan bir problemde optimal istasyon sayısından bir fazla sayıda istasyondan oluşan çözüm elde edilmiştir.

Tablo 4. 2 Önerilen BBAOA için U-Tipi MHD sonuçları

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Optimal İst. Sayısı	BBAOA Sonucu	Amaç Fonksiyonu	Parametreler				CPU (sn)
					S	N_k	$N_{ür}$	N_{ed}	
Merten (7)	6	6	6	33	2	2	0	0	0,01
	7	5	5	33	2	2	0	0	0,02
	8	5	5	37	2	2	0	0	0,01
	10	3	3	29	2	2	0	0	0,01
	15	2	2	29	2	2	0	0	0,01
	18	2	2	29	2	2	0	0	0,01
Bowman (8)	20	4	4	77	2	2	0	0	0,02
Jaeschke (9)	6	8	8	46	2	2	0	0	0,02
	7	7	7	46	2	2	0	0	0,02
	8	6	6	44	2	2	0	0	0,02
	10	4	4	40	2	2	0	0	0,02
	18	3	3	40	2	2	0	0	0,02
Mansoor (11)	48	4	4	186	2	2	0	0	0,03
	62	3	3	186	2	4	0	0	0,05
	94	2	2	185	2	2	0	0	0,03
Jackson (11)	7	7	7	47	2	4	0	0	0,04
	9	6	6	51	2	2	0	0	0,03
	10	5	5	46	2	2	0	0	0,04
	13	4	4	47	2	2	0	0	0,05
	14	4	4	47	2	2	0	0	0,03
	21	3	3	47	2	2	0	0	0,04
Mitchell (21)	15	8	8	110	2	2	0	0	0,13
	14	8	8	110	2	2	0	0	0,12
	21	5	5	105	2	2	0	0	0,18
	26	5	5	107	2	2	0	0	0,09
	35	3	3	105	2	2	0	0	0,14
	39	3	3	107	2	2	0	0	0,19
Heskia (28)	138	8	8	1073	2	2	0	0	0,35
	205	5	5	1024	4	10	0	0	1,63
	216	5	5	1043	2	2	0	0	0,28
	256	4	4	1024	2	4	0	0	0,37
	324	4	4	1025	2	2	0	0	0,26
	342	3	3	1025	2	2	0	0	0,31
Sawyer (30)	25	14	14	341	2	2	0	0	0,28
	27	13	13	351	2	2	0	0	0,41
	30	11	11	330	10	12	1	0	10,39
	36	9	10*	345	2	2	0	0	0,24
	41	8	8	327	2	4	0	0	0,59
	54	6	6	324	10	14	1	0	12,14
	75	5	5	328	2	2	0	0	0,35
K. & Wester (45)	57	10	10	568	2	2	0	0	1,15
	79	7	7	552	8	10	0	0	12,20
	92	6	6	552	8	10	0	0	12,51
	110	6	6	557	2	2	0	0	1,14
	138	4	4	552	2	4	0	0	1,61
	184	3	3	552	2	2	0	0	1,00

Tablo 4. 2 (Devam)

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Optimal İst. Sayısı	BBAOA Sonucu	Amaç Fonksiyonu	Parametreler				CPU (sn)
					S	N_k	$N_{ür}$	N_{ed}	
Tonge (70)	176	[20,21]	21	3695	2	2	0	0	3,71
	364	10	10	3583	2	2	0	0	3,56
	410	9	9	3555	2	2	0	0	3,66
	468	8	8	3565	2	2	0	0	3,91
	527	7	7	3562	2	2	0	0	3,66
Arcus (83)	5048	16	16	78610	2	2	0	0	5,81
	5853	13	14*	77823	2	2	0	0	5,13
	6842	12	12	77574	2	2	0	0	5,16
	7571	11	11	78286	2	2	0	0	4,36
	8412	10	10	77731	2	2	0	0	5,20
	8898	9	9	76964	2	2	0	0	5,18
	10816	7	8*	76868	2	2	0	0	5,83
Arcus (111)	5755	27	28*	160589	2	2	0	0	15,52
	8847	[17,18]	18	157706	2	2	0	0	14,43
	10027	[15,16]	16	155605	2	2	0	0	13,08
	10743	[14,15]	15	156135	2	2	0	0	17,04
	11378	14	14	155571	2	2	0	0	11,27
	17067	9	9	151435	2	2	0	0	14,94

* En yüksek parametre seviyelerine ulaşıncaya kadar deneme yapılmış ancak optimal istasyon sayısına ulaşılamamıştır. Optimalden bir fazla istasyon sayısıyla elde edilen çözümler için en düşük parametre seviyeleri ve ilgili çözüm süreleri gösterilmiştir.

Tablo 4.2'de önerilen BBAOA'nın U-Tipi MHD problemi sonuçları ve ilgili çözümlere ulaşıldığı andaki parametre seviyeleri ile çözüm süreleri verilmiştir. Problem adının yanında parantez içindeki sayı problemin görev sayısını ifade ederken, optimal sonuçların verildiği sütunda köşeli parantez kullanılması ilgili problemin optimal sonucunun bilinmediğini göstermektedir. Köşeli parantez içindeki sayılar optimal sonuç için alt ve üst sınırları ifade etmektedir. Tablo incelendiğinde, önerilen BBAOA'nın ilgili test problemleri üzerinde oldukça etkili olduğu, optimal sonuçlara çok kısa süreler içerisinde ulaştığı görülmektedir. Önerilen BBAOA ile Tablo 4.2'de verilen 64 adet test probleminin 60'ında optimal sonuçlara saniyelerle ifade edilen sürelerde ulaşılmış, yalnızca dört problemde, daha önce belirtildiği üzere problemin görev sayısına bağlı olarak ortaya çıkan en yüksek parametre seviyelerine kadar çözümler yapılmasına rağmen optimal sonuçlara ulaşılamamıştır. Önerilen BBAOA ile, U-Tipi MHD problemleri için optimal istasyon sayısına ulaşılamayan dört problemde de optimal istasyon sayısından bir fazla sayıda istasyondan oluşan çözümler elde edilmiştir.

Tablo 4.3'de düz ve U-Tipi MHD problemleri için, önerilen BBAOA ile Baykasoğlu ve Dereli (2009) tarafından önerilen KKOA sonuçlarının karşılaştırması; U-Tipi MHD problemleri için ise Ajenblit ve Wainwright (1998) tarafından önerilen GA sonuçlarının karşılaştırması verilmiştir.

Tablo 4. 3 Düz ve U-Tipi MHD problemleri için BBAOA, KKOA ve GA karşılaştırması

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Düz MHD			U-Tipi MHD			
		Optimal İst. Sayısı	BBAOA Sonucu	KKOA Sonucu	Optimal İst. Sayısı	BBAOA Sonucu	KKOA Sonucu	GA Sonucu
Metren (7)	6	6	6	6	6	6	6	6
	7	5	5	5	5	5	5	5
	8	5	5	5	5	5	5	5
	10	3	3	3	3	3	3	3
	15	2	2	2	2	2	2	2
	18	2	2	2	2	2	2	2
Bowman (8)	20	5	5	5	4	4	4	4
Jaeschke (9)	6	8	8	8	8	8	8	8
	7	7	7	7	7	7	7	7
	8	6	6	6	6	6	6	6
	10	4	4	4	4	4	4	4
	18	3	3	3	3	3	3	3
Mansoor (11)	48	4	4	4	4	4	4	4
	62	3	3	3	3	3	3	3
	94	2	2	2	2	2	2	2
Jackson (11)	7	8	8	8	7	7	7	7
	9	6	6	6	6	6	6	6
	10	5	5	5	5	5	5	5
	13	4	4	4	4	4	4	4
	14	4	4	4	4	4	4	4
	21	3	3	3	3	3	3	3
Mitchell (21)	14	8	8	8	8	8	8	8
	15	8	8	8	8	8	8	8
	21	5	5	5	5	5	5	5
	26	5	5	5	5	5	5	-
	35	3	3	3	3	3	3	-
	39	3	3	3	3	3	3	-
Heskia (28)	138	8	8	8	8	8	8	8
	205	5	5	5	5	5	5	6
	216	5	5	5	5	5	5	5
	256	4	4	4	4	4	4	4
	324	4	4	4	4	4	4	4
	342	3	3	3	3	3	3	3
Sawyer (30)	25	14	14	14	14	14	14	14
	27	13	13	13	13	13	13	13
	30	12	12	12	11	11	12	12
	36	10	10	10	9	10	10	10
	41	8	8	8	8	8	8	8
	54	7	7	7	6	6	7	7
	75	5	5	5	5	5	5	5

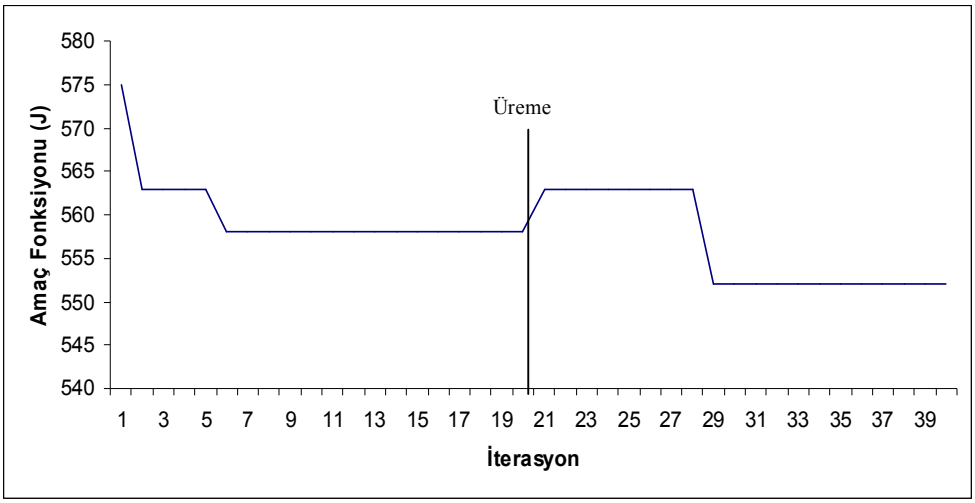
Tablo 4. 3 (Devam)

Problem Adı (Görev Sayısı)	Çevrim Zamanı	Düz MHD			U-Tipi MHD			
		Optimal İst. Sayısı	BBAOA Sonucu	KKOA Sonucu	Optimal İst. Sayısı	BBAOA Sonucu	KKOA Sonucu	GA Sonucu
K. & Wester (45)	57	10	10	10	10	10	10	10
	79	7	7	7	7	7	7	8
	92	6	6	6	6	6	6	6
	110	6	6	6	6	6	6	6
	138	4	4	4	4	4	4	4
	184	3	3	3	3	3	3	3
Tonge (70)	176	21	21	21	[20,21]	21	21	21
	364	10	10	10	10	10	10	10
	410	9	9	9	9	9	9	9
	468	8	8	8	8	8	8	8
	527	7	7	7	7	7	7	7
Arcus (83)	5048	16	16	16	16	16	16	16
	5853	14	14	14	13	14	14	14
	6842	12	12	12	12	12	12	12
	7571	11	11	11	11	11	11	11
	8412	10	10	10	10	10	10	10
	8898	9	9	9	9	9	9	9
	10816	8	8	8	7	8	8	8
Arcus (111)	5755	27	28	27	27	28	27	28
	8847	18	18	18	[17,18]	18	18	18
	10027	16	16	16	[15,16]	16	16	16
	10743	15	15	15	[14,15]	15	15	15
	11378	14	14	14	14	14	14	14
	17067	9	9	9	9	9	9	9

Tablo 4.3'de verilen, düz ve U-Tipi MHD problemleri için BBAOA ve KKOA karşılaştırmasında, düz MHD problemleri için önerilen BBAOA yalnızca bir problem haricinde optimal sonuçlara ulaşırken KKOA ile bütün problemlerde optimal sonuçların elde edildiği görülmektedir. U-Tipi MHD problemlerinde ise, iki problemde, önerilen BBAOA, KKOA'ya göre daha iyi sonuçlar elde ederken, bir problemde KKOA önerilen BBAOA'dan daha iyi sonuç elde etmiştir. Öte yandan, çalışma kapsamında ele alınan 64 U-Tipi MHD probleminden 61 adedinin Ajenblit ve Wainwright (1998) tarafından önerilen GA yaklaşımıyla çözümleri incelendiğinde; önerilen BBAOA'nın dört problemde GA'dan daha iyi sonuçlara ulaştığı görülmektedir. Bu veriler ışığında, Talbot ve ark. (1986) problem seti için her üç algoritmanın da oldukça yüksek ve birbirleriyle rekabet edebilecek düzeyde performans gösterdiği söylenebilir.

Önerilen BBAOA, MHD problemleri üzerinde literatürde ilk defa uygulanmış ve görev sayıları 7 ile 111 arasında değişen basit düz ve U-Tipi MHD problemlerinin

büyük çoğunluğunda optimal sonuçlara saniyelerle ifade edilebilen sürelerde ulaşmıştır. Algoritmanın optimal sonuçlara ulaşamadığı az sayıda problem, görev süreleri, öncelik ilişkileri ve çevrim zamanları itibariyle diğer problemlere nazaran daha zor problemlerdir. Bu problemlerde, optimal çözümde genellikle bütün istasyonların dolu olması ve toplam boş zamanın sıfıra yakın olması nedeniyle sezgisel ve metasezgisel yöntemlerle optimal çözümünün elde edilmesi diğer problemlere nazaran daha zor olabilmektedir.



Şekil 4. 1 K. & Wester problemi için optimal sonucu veren bir bakteriye ait optimizasyon süreci

Şekil 4.1'de, K. & Wester probleminin *C*=79 çevrim zamanı için düz MHD problemi olarak çözümünde optimal sonucu veren bir bakteriye ait optimizasyon süreci gösterilmiştir. Önerilen BBAOA, araştırma kapsamında çözülen problemlerin büyük bir kısmında çok düşük parametre seviyelerinde optimal sonuçlara ulaşmıştır. Düşük parametre seviyelerinde optimal sonuçlara ulaşılamadığı durumlarda ise parametrelerin artırılarak optimal sonuçlara ulaşılmış olması, algoritmanın geçerliğinin ve etkinliğinin bir göstergesidir.

5. SONUÇ VE ÖNERİLER

Günümüzde birçok sektörde yaygın olarak kullanılan montaj hatlarının performansı, üretim süreçlerinin son aşamaları olmaları itibariyle, tüm sistemin genel performansı üzerinde oldukça etkilidir. Son yıllarda, müşteri tercihlerinin son derece farklılaşması ve müşterilerin kendilerine özgü ürünler talep etmelerinin yanı sıra, teknolojinin de gün geçtikçe gelişmesiyle rekabet koşulları oldukça artmıştır. Bu rekabet koşulları altında, işletmeler, verimli, etkili ve esnek bir üretim sistemi kullanarak kaliteden ödün vermeden müşteri taleplerini çok hızlı bir şekilde karşılamak durumunda kalmışlardır. Son 50 yıldır, işletmelerin bu başarıyı yakalayabilmelerini sağlayacak en etkili üretim sistemi Toyota Üretim Sistemi olarak karşımıza çıkmaktadır. Toyota Üretim Sisteminin stoksuz üretim ve tek parça akışı gibi temel ilkeleri uygulamaya koyulduğunda montaj hatlarının önemi bir kat daha artmakta, diğer üretim süreçlerinin planlanması son montaj hattının malzeme ihtiyacına göre yapılmaktadır. Toyota Üretim Sistemini başarıyla uygulayabilmek için ise tesis yerleşiminin U-Tipi hatlarla oluşturulması gerekmektedir. Bu durum göz önüne alındığında, işletmelerin karşısına çıkabilecek en önemli karar problemleri arasında basit düz ve U-Tipi MHD problemleri gösterilebilir.

Bu çalışmada, MHD konsepti ve MHD problemleriyle ilgili genel bilgilerin yanı sıra, literatürde düz ve U-Tipi MHD problemlerine henüz uygulanmamış, yeni sayılabilecek bir metasezgisel yöntem olan BBAOA, ilgili problemlere uygulanmıştır. Toyota Üretim Sisteminin en önemli bileşenlerinden biri olan Shojinka prensibinin temelinde, talepteki ani bir değişikliğe karşı üretim birimini oluşturan hatların çok çabuk bir şekilde yeninden dengelenmesi yeteneği yattığından dolayı, NP-Zor sınıfına giren bu problemin, kabul edilebilir bir süre içerisinde tatmin edici çözümlere kavuşturulması gerekmektedir. Bu nedenle, MHD literatüründe daha önceden önerilmiş performansı test edilmiş yöntemlere ilave olarak yeni geliştirilen meta sezgisel yöntemlerin, probleme uyarlanarak literatürün geliştirilmesi son derece önemlidir.

Çalışma kapsamında, MHD problemlerine ilk kez uygulanan BBAOA ile düz ve U-Tipi MHD problemleri için etkili sonuçlar elde edilmiş, 128 test probleminin 123'ünde kabul edilebilir süreler içerisinde optimal sonuçlara ulaşılmıştır. Önerilen BBAOA'nın MHD problemleri üzerinde, karşılaştırma yapılan KKOA ve GA ile rekabet edebilecek düzeyde performans gösterdiği görülmüş, gelecekte daha da geliştirilerek MHD problemlerinin diğer varyasyonları üzerinde de başarıyla uygulanabileceği konusunda fikir sahibi olunmuştur. Bu çalışmada, en yalın hali ve temel ilkelerine birebir bağlı kalınarak uygulanan BBAOA'nın, özellikle algoritmanın çekirdeği niteliğini taşıyan kemotaksis süreçte ve başlangıç çözümlerinde yapılacak iyileştirmelerle, daha büyük boyutlu ve nispeten daha zor MHD problemleri üzerinde de başarıyla uygulanabileceği öngörüsüne ulaşılmıştır. Başlangıç çözümlerinde yapılacak iyileştirmeler, bakterilerin yaşamlarına besince zengin bölgelerde başlamasına, bir başka deyişle optimizasyon süreçlerine en iyi çözüme daha yakın çözümlerle başlamalarına ve mümkün olan en iyi çözümleri çok daha kısa sürelerde bulmalarına katkı sağlayacaktır.

Ayrıca, büyük boyutlu ve daha zor problemlerde kısa sürede etkin çözümler elde etmek amacıyla, BBAOA ile literatürde bilinen diğer metasezgisel yöntemler birleştirilerek melez algoritmalar geliştirilmesi yoluyla da etkin çözümlere ulaşılabileceği söylenebilir.

6. KAYNAKLAR

Aase, G.R., 1999. Algorithms for Increasing Labor Productivity in U-Shaped Assembly Systems. Yayınlanmamış Doktora Tezi, Indiana Üniversitesi, ABD.

Aase, G.R., Marc, J.S., John, R.O., 2003. U-OPT: an analysis of exact U-shaped line balancing procedures. International Journal of Production Research 41: 17, 4185–4210.

Aase, G.R., John, R.O., Marc, J.S., 2004. U-shaped assembly line layouts and their impact on labor productivity: An experimental study. European Journal of Operational Research 156:3, 698–711.

Ağpak, K., Gökçen, H., 2001. U-Tipi Montaj Hattı Dengelenmesi için Bir Sezgisel Metot: Düzenlenmiş COMSOAL (U-COMSOAL). Endüstri Mühendisliği Dergisi 12: 2, 23-32.

Ağpak, K., Gökçen, H., 2002. Basit U Tipi Montaj Hattı Dengeleme Problemine Bulanık Programlama Yaklaşımı. DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi 4: 2, 29-40.

Ajenblit, D.A., Wainwright R.L., 1998. Applying Genetic Algorithms to the U-Shaped Assembly Line Balancing Problem. Proceedings of the IEEE International Conference on Evolutionary Computation: 96-101.

Anonim 1, 2010. http://www.assembly-line-balancing.de.

Başbuğ, S., 2008. Bakteriyel Besin Arama Algoritması ile Lineer Anten Dizilerinin Diyagram Sıfırlaması. Yayınlanmamış Yüksek Lisans Tezi, Erciyes Üniversitesi, Kayseri.

Baybars, I., 1986. A Survey of Exact Algorithms for the Simple Line Balancing Problem. Management Science 32: 909-932

Baykasoğlu, A., Dereli, T., 2009. Simple and U-Type Assembly Line Balancing by Using an Ant Colony Based Algorithm. Mathematical and Computational Applications 14:1, 1-12.

Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 168: 694-715.

Biswas, A.S., Dasgupta, S., Das, A., Abraham, A. 2007a. Synergy of pso and Bacterial foraging optimization - a comparative study on numerical benchmarks. In E. C. et al., editor, Innovations in Hybrid Intelligent Systems, Spronger-Verlag, 255– 263.

Biswas, A.S., Dasgupta, S., Das, S., Abraham A. 2007b. A synergy of differential evolution and bacterial foraging algorithm for global optimization. International Journal on Neural and Mass-Parallel Computing and Information Systems, Neural Network World 17: 6, 607-626.

Bremermann, H..1974. Chemotaxis and optimization, J. Franklin Inst 297, 397–404.

Cheng,C.H., Miltenburg, J., Motwani, J., 2000. The Effect of straight and U-shaped lines on quality. IEEE Transactions on Engineering Management 47:3,

Chiang, W.C., Kouvelis, P., Urban, T.L., 2007. Line balancing in a just-in-time production environment: balancing multiple U-lines. IIE Transactions 39: 347-359.

Deshpande, V., Rajaram, K., Guignard, M., 1997. An Integer Programming Based Formulation of the U-Line Balancing Problem. INFORMS Meeting, San Diego, ABD

Erel, E., Sarin, S.C., 1998. A Survey of the Assembly Line Balancing Procedures. Production Planning & Control, 9:, 5, 414-434.

Erel, E., Sabuncuoğlu, I., Aksu, B.A., 2001. Balancing of U-Type Assembly Systems Using Simulated Annealing. International Journal of Production Research 39: 13, 3003-3015

Ghosh, S., Gagnon, J., 1989. A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems. International Journal of Production Research 27: 637-670

Gökçen, H., Ağpak, K., Benzer, R., 2006. Balancing Parallel Assembly Lines. International Journal of Production Economics 103: 600-609.

Gökçen, H., Ağpak, K., 2006. A Goal Programming Approach to Simple U-Line Balancing Problem. European Journal of Operational Research 171: 2, 577-585.

Hwang, H., Sun, J.U., Yoon T.H., 1996. U-Line Line Balancing with Simulated Annealing. First Asia Pacific Decision Sciences Institute Conference, Hong Kong, June 21-22, 101-108

Hwang, R.K., Katayama, H., Gen, M., 2008. U-shaped assembly line with genetic algortihm. International Journal of Production Research 46:16, 4637-4649.

Hwang, R.K., Katayama, H., 2010. Uniform workload assignments for assembly line by GA-based amelioration approach International Journal of Production Research 48:7, 1857-1871.

Kara, Y., 2004. U-Tipi Montaj Hattı Dengeleme Problemleri İçin Yeni Modeller Ve Otomotiv Yan Sanayiinde Bir Uygulama. Yayınlanmamış Doktora Tezi, Selçuk Üniversitesi, Konya.

Kara, Y., Özcan, U., Peker, A., 2007a. An approach for balancing and sequencing mixed-model JIT U-lines. The International Journal of Advanced Manufacturing Technology 32:11-12, 1218-1231.

Kara, Y., Özcan, U., Peker, A., 2007b. Balancing and sequencing mixed-model just-in-time U-lines with multiple objectives. Applied Mathematics and Computation 184:2, 566-588 .

Kara, Y., 2008. Line balancing and model sequencing to reduce work overload in mixed-model U-line production environments. Engineering Optimization 40:7, 669-684.

Kara, Y., Tekin, M., 2009. A mixed integer linear programming formulation for optimal balancing of mixed-model U-lines. International Journal of Production Research 47:15, 4201-4233.

Kara, Y., Paksoy, T., Chang, C.T. 2009. Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing. European Journal of Operational Research 195:2, 335–347.

Kim, Y.K., Kim, S.J., Kim, J.Y., 2000. Balancing and Sequencing Mixed-Model U-Lines with a Co-Evolutionary Algorithm. Production Planning & Control 11:8, 754-764.

Kim, D.H., Cho, J.H. 2005a. Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization, in: Piotr S. Szczepaniak, Janusz Kacprzyk, Adam Niewiadomski (Eds.), Third International Atlantic Web Intelligence Conference,AWIC. Lodz, Poland, Lecture Notes in Computer Science. 3528, 231–238.

Kim, D.H., Cho C.H. 2005b. Bacterial foraging based neural network fuzzy learning. Proceedings of 1st Indian International Conference on Artificial Intelligence. 2030-2036,

Kim, D.H., Abraham, A., Cho, J.H. 2007. A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences 177:18, 3918-3937.

Majhi, B., Panda, G.. 2007. Bacteria foraging based identificacion of nonlinear dynamic system. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, IEEE Service Center, 1636–1641.

Majhi, B., Panda, G., Sahoo, G., Dash, P., Das D. 2007. Stock market prediction of s&p 500 and djia using bacterial foraging optimization technique. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2007), Singapore, IEEE Service Center, September: 2569–2575.

Merten, P., 1967. Assembly Line Balancing by Partial Enumeration. Ablaufund Planungforschung 8, 429-433.

Miltenburg, J., Wijngaard, J., 1994. The U-Line Line Balancing Problem, Management Science 40: 1378-1388

Miltenburg, J., 1998. Balancing U-Lines in a Multiple U-Line Facility, European Journal of Operational Research 109: 1-23

Miltenburg, J., 2000. The effect of breakdowns on U-shaped production lines. International Journal of Production Research 38:2, 353 – 364.

Miltenburg, J., 2001a. U-shaped production lines: A review of theory and practice. International Journal of Production Economics 70: 201–214.

Miltenburg, J. 2001b. One-piece flow manufacturing on U-shaped production lines: a tutorial. IIE Transactions 33:4, 303-321.

Miltenburg, J., 2002. Balancing and Scheduling Mixed Model U-Shaped Production Lines. The International Journal of Flexible Manufacturing Systems 14, 119-151.

Mishra, S., 2005. A hybrid least square-fuzzy bacteria foraging strategy for harmonic estimation. IEEE Trans. Evol. Comput. 9: 1, 61–73.

Mishra, S., Reddy, G. D., Rao, P. E., Santosh, K.. 2007. Implementation of new evolutionary techniques for transmission loss reduction. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, IEEE Service Center, 2331– 336.

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine, 52–67.

Sabuncuoğlu, İ., Erel, E., Alp, A., 2009. Ant colony optimization for the single model U-type assembly line balancing problem. International Journal of Production Economics 120, 287-300.

Salveson, M.E., 1955. The Assembly Line Balancing Problem. Journal of Industrial Engineering 6: 3, 18-25

Scholl, A., ve Klein, R., 1999. ULINO: Optimally Balancing U-Shaped JIT Assembly Lines. International Journal of Production Research 37: 4, 721-736

Sparling, D., 1997. Topics in U-Line Balancing. Yayınlanmamış Doktora Tezi, McMaster Üniversitesi, Kanada

Sparling, D., 1998. Balancing Just-In-Time Production Units: The N U-Line Balancing Problem. INFOR 36: 1, 215-237

Sparling D., Miltenburg, J., 1998. The mixed-model U-line balancing problem. International Journal of Production Research 36:2 485-501.

Suwannarongsri, S., Puangdownreong, D. 2008. Balancing of U-Shape Assembly Lines using Tabu Search Method. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008. 5th International Conference on Proceedings: 609-612.

Talbot, F.B., ve Patterson, J.H., 1984. An Integer Programming Algorithm with Network Cuts for Solving the Single Model Assembly Line Balancing Problem. Management Science 30: 85-99

Talbot, F.B., Patterson, J.H., Gerhlein, W.V., 1986. A comparative evaluation of heuristic line balancing techniques. Management Science 32:4,430-454.

Tripathy, M., Mishra, S., Lai, L. L., Zhang,Q.P., 2006. Transmission loss reduction based on FACTS and Bacteria Foraging Algorithm. Proceedings of 9th International Conference on Parallel Problem Solving from Nature:. 222-231.

Urban, T.L., 1998. Optimal Balancing of U-Shaped Assembly Lines. Management Science 44: 5, 738-741.

Wu, C., Zhang, N., Jiang, J., Yang, J., Liang, Y., 2007. Improved Bacterial Foraging Algorithms and Their Applications to Job Shop Scheduling Problems. ICANNGA 2007, Part I, LNCS 4431, 562 – 569.