



İş yükü dengelemeli ikinci tip basit montaj hattı dengeleme problemi için bir tabu arama algoritması

Murat Arıkan*

Gazi Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 06570, Maltepe, Ankara, Türkiye

Ö N E Ç İ K A N L A R

- İki amaçlı ikinci tip basit montaj hattı dengeleme problemi ele alındı
- Çözüm için göreve yönelik çözüm temsili kullanan, kalıcılık sıklığına dayalı bir çeşitlendirme stratejisi ile desteklenmiş bir tabu arama algoritması geliştirildi
- Algoritmanın performansı açık literatürden alınan test problemleri üzerinde değerlendirildi

Makale Bilgileri

Geliş: 19.07.2016

Kabul: 05.11.2016

DOI:

10.17341/gazimmfd.369529

ÖZET

Bu çalışmada, iki amaçlı tip-2 basit montaj hattı dengeleme probleminin çözümü için bir tabu arama algoritması geliştirilmiştir. Algoritma, aynı problem için genellikle evrimsel algoritmalarda yararlanılan göreve yönelik bir çözüm temsili kullanmakta ve kalıcılık sıklığına dayalı bir çeşitlendirme stratejisiyle desteklenmektedir. Algoritmanın performansı, açık literatürden alınmış 9 probleme ait 23 örnek üzerinde test edilmiştir. Hesaplama sonuçları, her iki amaç için de algoritmanın etkinliğini doğrulamaktadır.

Anahtar Kelimeler:

Montaj hattı dengeleme,
tip-2 basit montaj hattı
dengeleme problemi,
tabu arama

A tabu search algorithm for the simple assembly line balancing problem of type-2 with workload balancing objective

H I G H L I G H T S

- The bi-objective type-2 simple assembly line balancing problem is considered
- To solve the problem, a tabu search algorithm is developed that uses a task oriented solution representation and is supported by a diversification strategy based on residence frequencies
- The performance of the algorithm is evaluated on test problems taken from the open literature

Article Info

Received: 19.07.2016

Accepted: 05.11.2016

DOI:

10.17341/gazimmfd.369529

ABSTRACT

In this study, a tabu search algorithm is developed to solve the bi-objective simple assembly line balancing problem of type-2. The algorithm uses a task oriented solution representation which is usually employed by evolutionary algorithms and is supported by a diversification strategy based on residence frequencies. The performance of the algorithm is tested on 23 instances included in 9 problems taken from the open literature. The computational results validate the effectiveness of the algorithm for both objectives.

Keywords:

Assembly line balancing,
type-2 simple assembly line
balancing problem,
tabu search

*Sorumlu Yazar/Corresponding Author: marikan@gazi.edu.tr / Tel: +90 312 582 3808

1. GİRİŞ (INTRODUCTION)

Montaj hatları, genellikle hareketli bant veya konveyör sistemiyle bağlanmış birbirini takip eden iş istasyonlarında alt parçaların, birleştirilerek ya da üzerlerinde birtakım işlemler gerçekleştirilerek, son ürünlere dönüştürüldüğü sistemlerdir. İşlenecek parçalar, önceden belirlenmiş sabit bir hızda ilk istasyondan hatta girer, her istasyonda belli işlemlere tabi tutularak son istasyondan hattı terk ederler. Montaj hatlarıyla ilgili temel problemlerden biri, belli bir performans ölçütünü en iyileyecek şekilde son ürünü oluşturan görevlerin ya da operasyonların iş istasyonlarına atanması olarak tanımlanan montaj hattı dengeleme problemi. Bu görevler, iş istasyonlarına atanırken birtakım kısıtları sağlamak durumundadır. Bunların ilki, her bir istasyona atanan görevlerin toplam zamanının çevrim süresini geçmeme zorunluluğudur. Çevrim süresi, belli bir zaman periyodu içinde kaç ürün üretilmesi gerektiğini ifade eden hattın üretim hızı tarafından belirlenir. İstenen üretim hızını tutturabilmek için her çevrim süresinde bir ürün hattan çıkmalıdır. İkinci kısıt, atamanın görevlerin öncelik sıralarını ihlal etmeme şartıdır. Bir görevin bir istasyona atanabilmesi için öncüllerinin (kendisinden önce gelen görevlerin) atanmış olması gerekir. Diğer bir kısıt ise her görevin sadece tek bir istasyona atanabileceğidir. NP-zor olduğu bilinen basit montaj hattı dengeleme probleminin, literatürde sıklıkla ele alınan, tip-1 ve tip-2 olmak üzere iki temel çeşidi vardır. Tip-1 montaj hattı dengeleme problemlerinde çevrim süresi verilir, istasyon sayısının minimize edilmesi amaçlanır. Tip-2 problemlerde istasyon sayısı bellidir, çevrim süresinin minimize edilmesi hedeflenir. Üretim hızını sabit bir parametre olarak belirlemek gerektiğinden yeni bir montaj hattının kurulmasına karar verildiğinde ve ürün talebi iyi tahmin edilebildiğinde tip-1 problemin çözümüne ihtiyaç vardır. Bir montaj hattı mevcutken, üretim süresinde ya da talep yapısında değişiklikler olduğunda ve üretim hızını arttırmak gerektiğinde ise tip-2 problem çözümlenmelidir [1]. Tanımlarından da anlaşılacağı gibi bu iki problemin standart yapısında sadece bir amaç dikkate alınmaktadır. Ancak, montaj hattı dengeleme problemlerinin tek bir amaç kullanılarak çözülmesi, istasyonlar arasında dengesiz bir görev dağılımına yol açabilmektedir. Her istasyona kıyaslanabilir iş, dolayısıyla her işçiye yaklaşık boş zaman atamaya çalışmak yaygın bir uygulamadır. Aksi halde, işçiler arasında adaletsiz çalışma koşullarına neden olmak kaçınılmazdır [2]. Basit montaj hattı dengeleme problemi üzerine yapılan çalışmalar incelendiğinde, literatürün büyük bir bölümünün tip-1 probleme odaklandığı görülmektedir. İlk zamanlarda, birçok araştırmacı tip-2 montaj hattı dengeleme problemlerini tekrarlı olarak tip-1 problemi çözen yaklaşımlarla ele almıştır. Bu yaklaşımlarda, farklı deneme çevrim süreleri sırayla dikkate alınarak, sayısı belirli istasyonlara tüm görevlerin uygun bir atamasının olup olmadığı kontrol edilir. Böyle bir teknik kullanıldığında, çevrim süresi için alt sınır ve/veya üst sınır değerleri hesaplanır ve genellikle, çevrim süresini, Mastor [3] ve Gehrlin ve Patterson [4] gibi alt sınırdan başlayıp belli oranlarda arttırarak ya da Hackman vd. [5] gibi alt sınır ve

üst sınır değerleri arasında belli kurallara göre güncelleyerek tekrarlı bir şekilde tip-1 problemi çözen stratejilerle çözüme ulaşılır. Daha sonraları, tip-2 montaj hattı dengeleme problemlerini doğrudan çözen yaklaşımların da önerildiği görülmektedir. Klein ve Scholl [6], SALOME-1'in bir uyarlaması olan ve yeni bir sayma tekniği kullanan dal-sınır yöntemi, SALOME-2'yi geliştirmişlerdir. Uğurdağ vd. [7], tip-2 montaj hattı dengeleme problemi için tam sayılı programlamaya dayanan iki aşamalı bir sezgisel yöntem önermişlerdir. İlk aşamada, geliştirilen sezgisel bir prosedür ile uygun bir çözüm bulunmakta, ikinci aşamada ise bulunan başlangıç çözümü simplekse benzer bir algoritmayla iyileştirilmektedir. Liu vd. [8], ilk önce iki yönlü bir atama prosedürüyle başlangıç çözümünü elde eden, daha sonra da iş istasyonları arasında işleri değiştirerek bu çözümü iyileştiren iki sezgisel algoritma önermişlerdir. Kılınççı [9], petri ağları tabanlı bir sezgisel, Blum [10] ise tekrarlı bir ışın arama algoritması sunmuştur. La Scalia vd. [11], problem bulanık işlem zamanlarına sahip olduğunda kullanılabilir bir bulanık ikili doğrusal programlama modeli ve çözüm algoritması önermişlerdir. Bunlara ek olarak, meta-sezgisel yaklaşımların NP-zor problemlerin çözümündeki başarısının takdir edilmesiyle birlikte, tip-2 montaj hattı dengeleme problemlerinin bu yöntemlerle ele alındığı çalışmalara da rastlanmaya başlamıştır. Heinrici [12], tip-2 problem için basit birer tavlama benzetimi (TB) ve tabu arama algoritması (TA) geliştirmiş ve performanslarını açık literatürden alınan bazı test problemleri üzerinde karşılaştırmıştır. Scholl ve Voß [1], birçok seçenekli elemana sahip bir tabu arama algoritması sunmuş ve geliştirdikleri prosedürün kurucu yöntemlere göre çok daha iyi sonuçlar verdiğini göstermiştir. Kim vd. [13], eş zamanlı olarak iş yükü düzgünlüğü ve iş benzerliği amaçlarını da dikkate alan bir çok amaçlı genetik algoritma (GA) geliştirmiştir. Nearchou, amaçları, çevrim süresi minimizasyonu [14] ve çevrim süresi ile iş istasyonlarının denge gecikme zamanı/hat iş yükünün düzgünlük indeksi minimizasyonu olan [15] diferansiyel evrim algoritmaları (DEA) sunmuş ve performanslarını daha önce geliştirilmiş iki genetik algoritma ile karşılaştırmıştır.

Nearchou [16], Parçacık Sürü Optimizasyonuna (PSO) dayanan yeni bir yöntem sunmuştur. Yöntemde, çevrim süresinin minimizasyonu ve iş yükü düzgünlüğünün maksimizasyonu gibi iki kriter eş zamanlı olarak dikkate alınmıştır. Algoritma ve iki mevcut çok amaçlı popülasyon sezgiseli arasında yapılan karşılaştırmalar, önerilen yaklaşımın ümit verici yüksek bir performansa sahip olduğunu göstermiştir. Zacharia ve Nearchou [17], işlem zamanlarının kesin olmadığı durum için iki amaçlı bir genetik algoritma tasarlamışlardır. Zheng vd. [18], problemin çözümü için istasyon karınca kolonisi optimizasyonu adı verilen iyileştirilmiş bir karınca kolonisi optimizasyon algoritması önermiş, algoritmanın etkinliğini ve istikrarını dokuz probleme ait 23 örnekte literatürle karşılaştırarak doğrulamışlardır. Modzgir vd. [19], aynı problemde iş yükü düzgünlük göstergesini minimize etmeyi amaçlayan bir diferansiyel evrim algoritması geliştirmişlerdir. Zacharia vd. [20], bir robot kolu montaj

hattı için ikinci tip basit montaj hattı dengeleme problemini çözen bir genetik algoritma önermişlerdir. Zhang vd. [21] ise problemin çözümü için doğrudan tam sayı değişkenleri kullanabilen yeni bir diferansiyel evrim algoritması versiyonu geliştirmiş ve performansını literatürde mevcut DEA ve GA algoritmalarıyla karşılaştırmışlardır. Son yıllarda tip-2 montaj hattı dengeleme probleminin farklı tiplerinin de ele alındığı görülmektedir. Polat vd. [22], montaj hattı işçi atama ve dengeleme probleminin, Lei ve Guo [23] ise çift taraflı montaj hattı dengeleme probleminin ikinci tipi için değişken komşu arama algoritmaları geliştirmişlerdir. Rabbani vd. [24], U şekilli hatlarda tip-2 robotik karışık modellenmiş montaj hattı dengeleme probleminin çözümü için iki farklı çok amaçlı evrimsel algoritma önermiştir. Roshani ve Giglio [25], her bir istasyona birden fazla işçinin atanabildiği montaj hattı dengeleme problemini çevrim süresinin minimize edilmesi amacıyla ele almış ve iki tavlama benzetimi algoritması sunmuştur. Şahin ve Kellegöz [26] ise üretim oranını maksimize etme amaçlı U şekilli montaj hattı dengeleme probleminin çözümü için bir genetik algoritma önermiştir.

Bu çalışmada, iş yükü düzleştirilmesinin de beraberinde amaçlandığı ikinci tip basit montaj hattı dengeleme problemi için bir tabu arama algoritması geliştirilmiştir. Algoritmanın literatürde aynı problem için geliştirilmiş tabu arama algoritmalarından farkı, genellikle genetik [13] ve diferansiyel evrim [14] algoritmalarına benzer popülasyon sezgisellerinde olduğu gibi göreve yönelik bir çözüm temsili kullanmasıdır. Tip-2 basit montaj hattı dengeleme problemleri için geliştirilen tavlama benzetimi [12] ve tabu arama [1] gibi meta-sezgisel arama stratejilerinde ise hangi görevin hangi istasyona atanacağını belirten istasyona yönelik bir çözüm temsilinden yararlanılmıştır. Ayrıca, geliştirilen algorithma söz konusu çalışmalarda tavlama benzetimi ve tabu arama yöntemlerinden farklı olarak iki amaç dikkate alınmıştır. Algoritmanın etkinliği, açık literatürden alınan 9 probleme ait 23 örnek üzerinde, çevrim süresi amacı için Uğurdağ vd. [7]'nin sonuçları ve literatürden alınan optimal çözüm değerleriyle, iş yükü düzleştirme amacı için de problemlerin matematiksel modellerinin GAMS programının CPLEX çözücüsüyle çözümlerinden elde edilen sonuçlarla karşılaştırılarak test edilmiştir. Çalışmanın ikinci bölümünde, problemin tanımına ve matematiksel modeline yer verilmiştir. Geliştirilen tabu arama algoritmasından üçüncü bölümde, yapılan deneysel çalışma ve sonuçlarından dördüncü bölümde bahsedilmiştir. Çalışmanın özeti ve ileriki çalışmalar için öneriler ise beşinci bölümde yer almaktadır.

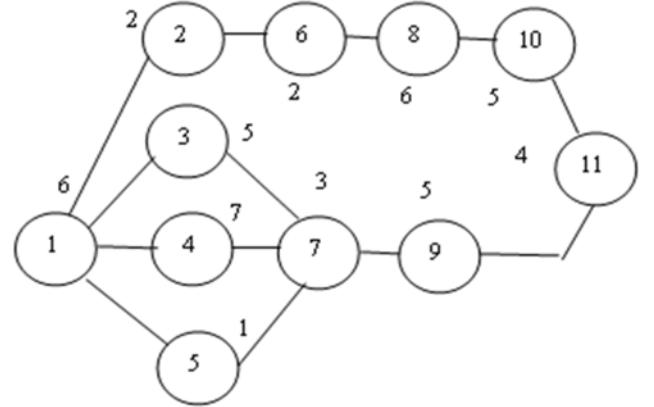
2. PROBLEMİN TANIMI VE MATEMATİKSEL MODELİ

(PROBLEM DESCRIPTION AND ITS MATHEMATICAL MODEL)

Basit montaj hattı dengeleme probleminde üretimi yapılan tek bir model ve o modele ait, birbirlerine göre öncelikleri bir öncelik diyagramıyla belirlenmiş, bir görev kümesi mevcuttur. Şekil 1'de örneği verilen öncelik diyagramı (Jackson, 1956) 11 görev içermektedir. Düğümün içindeki rakamlar görev numarasını, dışındaki rakamlar ise görev zamanını göstermektedir. Öncelik diyagramına göre, görev 3, 4 ve 5, görev 7 başlamadan bitirilmelidir, görev 3, 4 ve 5'in gerçekleştirilebilmesi için de görev 1'in tamamlanmış olması gerekmektedir. Bu çalışmada incelenen problem, ilk olarak çevrim süresinin, daha sonra da iş yükü dengesizliğinin minimize edilerek görevlerin iş istasyonlarına atanmasıyla ilgilenmektedir, bir başka deyişle iş yükü düzgünlüğünün ikinci kriter olarak dikkate alındığı tip-2 basit montaj hattı dengeleme problemidir. İş yükü düzgünlüğü için literatürde birçok farklı fonksiyon tanımlanmış [27], burada ise iş istasyonuna atanan görevlerin ortalama iş yükünden farklarının mutlak değerleri toplamı (Eş. 1) kullanılmıştır [2].

$$\sum_{j=1}^m \left| W_j - \frac{\sum_{i=1}^n t_i}{m} \right| \quad (1)$$

Bu formüldeki m istasyon sayısını, t_i i görevinin işlem zamanını, $\sum_{i=1}^n t_i$ tüm görevlerin işlem zamanları toplamını, W_j de j . iş istasyonuna atanan görevlerin işlem zamanları toplamını göstermektedir.



Şekil 1. Jackson (1956) probleminin öncelik diyagramı [28]
(Precedence graph of Jackson (1956) problem [28])

Tablo 1. Üç istasyonlu Jackson probleminde iki farklı görev ataması ve amaç fonksiyonları
(Two different task allocations and their objective functions in Jackson problem with three work stations)

İstasyonlar	Atama 1		Atama 2	
	Atanan Görevler	İstasyon iş yükü	Atanan Görevler	İstasyon iş yükü
1	1, 2, 3, 5, 6	16	1, 2, 4	15
2	4, 7, 8	16	3, 5, 6, 7, 9	16
3	9, 10, 11	14	8, 10, 11	15
Çevrim Süresi		16		16
Toplam dengesizlik		2,67		1,33

Şekil 1'deki öncelik diyagramına sahip bir ürünün üretildiği, 3 istasyonlu bir montaj hattında, her bir işçiye düşen ortalama iş yükü $\sum_{i=1}^n t_i/3$ formülünden 15,33 olarak hesaplanır ve adaletli bir iş yükü ataması yapmak için istasyonlara atanan görevlerin toplam zamanının ortalama iş yükünden mutlak sapmaları toplamı minimize edilmeye çalışılır. Tablo 1'de iki alternatif atama verilmiş ve atamalara ait çevrim süreleri ve toplam iş yükü dengesizlikleri hesaplanmıştır. İki atamanın da çevrim süresi 16 olmasına rağmen, ikincisinin toplam dengesizliğinin daha küçük olması istasyon, dolayısıyla işçi yüklerinin birbirine daha yakın olduğunu göstermektedir. Problemin varsayımları, matematiksel modeli ve kısıt açıklamaları aşağıda verilmiştir.

Varsayımlar:

- Her görevin işlem zamanı sabittir.
- Tüm görevler gerçekleştirilmelidir.
- Tüm iş istasyonları özdeşdir. Dolayısıyla, her görev her bir istasyonda, aynı işlem zamanıyla gerçekleştirilebilir.
- Montaj hattı, geleneksel (düz) yerleşime sahiptir.
- Hiç bir görev iki ya da daha fazla istasyon arasında bölünemez.
- İstasyonlara atanan görevler öncelik ilişkilerini ihlal edemez, yani bir görevin bir istasyona atanabilmesi için öncüllerinin aynı ya da daha önceki istasyonlara atanmış olması gerekir.

Eş. 2 hattın çevrim süresini en küçükleyen amaç fonksiyonudur.

$$\text{Min CT} \quad (2)$$

Eş. 3 ise istasyon iş yüklerinin ortalama iş yükünden aşağı ve yukarı sapmalarının toplamını minimize eden amaç fonksiyonudur.

$$\text{Min } \sum_{j=1}^m (O_j + U_j) \quad (3)$$

Eş. 4 her görevin bir istasyona atanmasını sağlamaktadır.

$$\sum_{j \in FS_i} x_{ij} = 1, \quad \forall i \quad (4)$$

Eş. 5 bir istasyona atanan görev zamanları toplamının çevrim süresini aşmamasını garanti etmektedir.

$$\sum_{i \in FT_j} t_i \cdot x_{ij} \leq CT, \quad \forall j \quad (5)$$

Eş. 6 görevler arasındaki öncelik ilişkilerinin ihlal edilmesini engellemektedir.

$$\sum_{j \in FT_j} j \cdot x_{ij} \leq \sum_{k \in FT_j} j \cdot x_{kj} \quad \forall (i, k) \in PR \quad (6)$$

Eş. 7 ile her istasyon için ortalama iş yükünden aşağı ve yukarı sapmalar belirlenmektedir.

$$\sum_{i \in FT_j} t_i \cdot x_{ij} + U_j - O_j = T/m \quad \forall j \quad (7)$$

Eş. 8, 9 ve 10 değişken tiplerini tanımlamaktadır.

$$x_{ij} \in \{0,1\} \quad \forall i, \forall j \in FS_i \quad (8)$$

$$CT \geq 0 \quad (9)$$

$$U_j, O_j \geq 0 \quad \forall j \quad (10)$$

Problem, ilk önce Eş. 2 amaç fonksiyonu altında çözülür ve montaj hattı için minimum çevrim süresi elde edilir. Daha sonra, ilk aşamada elde edilen çevrim süresi modelin altına kısıt olarak eklenir ve model, Eş. 3 amacı altında tekrar çözdürülür. Böylece çevrim süresi minimize edilirken işçiler arasında en dengeli iş yükü ataması da yapılmış olur.

Yukarıda matematiksel modeli verilen iki amaçlı tip-2 basit montaj hattı dengeleme probleminin çözümü için bir tabu arama algoritması geliştirilmiştir. Takip eden bölümde tabu arama hakkında bilgi verilmiş ve geliştirilen algoritmanın detayları anlatılmıştır.

3. ÖNERİLEN TABU ARAMA ALGORİTMASI (PROPOSED TABU SEARCH ALGORITHM)

Tabu Arama (TA), Glover [29] tarafından kombinatoriyal problemlerin çözümü için önerilmiş yüksek seviyeli bir sezgisel programlama tekniğidir. TA, yerel arama prosedürünü yerel optimumun ötesindeki çözüm uzayını araştırmaya yönlendiren bir sezgisel programlama tekniği olarak tanımlanmaktadır. TA'nın ayırtıcı özelliği hafızası yani, arama boyunca topladığı geçmiş kayıtlardan yararlanmasıdır. Yöntem, bir başlangıç çözümünden başlar ve iteratif bir şekilde daha iyi çözümlere ulaşmaya çalışır. Her iterasyonda, mevcut çözümün komşularının bir alt kümesi araştırılır ve daha kötü bir amaç fonksiyonu değerine sahip olsa da komşuların en iyisi mevcut çözüm olarak kaydedilir. Böylece, aramanın yerel optimumlara takılması engellenmeye çalışılır. Ancak, algoritma bu şekliyle döngüye girebilir. TA, aramayı yeni alanlara yönlendirebilmek ve tekrarlamasını önlemek için arama boyunca elde edilen geçmiş çözümlerin belli özelliklerini eski haline getiren hareketleri (çözümler arasındaki geçişleri) yasaklar. Yapılmasına izin verilmeyen hareketler "tabu" olarak sınıflandırılırlar ve "tabu listesi" adı verilen bir liste üzerinde kısa dönemli hafızada kaydedilirler. Bu hareketler belli bir süre sonra tabu listesinden çıkarılır ve tekrar yapılmalarına izin verilir. Bazı durumlarda bir hareket, tabu olmasına rağmen gerçekleştirilmek üzere seçilebilir. Bunun için dikkate alınan bir tabu yıkma kriterini sağlaması gerekmektedir. Birçok çalışmada kısa dönem hafıza yapısının yüksek kaliteli çözümlere ulaşmak için yeterli olduğu raporlanmış olmasına rağmen, TA, uzun dönem hafıza yapısını destekleyen çeşitlendirme ve yoğunlaştırma stratejilerinin kullanılması ile çok daha iyi sonuçlara ulaşabilmektedir [30]. Çeşitlendirme stratejileri aramayı, çözüm uzayının ziyaret edilmemiş bölgelerine yönlendirmeyi hedeflerken yoğunlaştırma stratejileri uzayın cezbedici bölgelerinin daha detaylı bir şekilde aranmasını teşvik eder. Tabu Aramayı kombinatoriyal bir probleme

uyarlamak için bazı kararların alınması gereklidir. Bu çalışmada, yöntemin iki amaçlı tip-2 montaj hattı dengeleme problemine uygulanması sırasında alınan kararlar aşağıda açıklanmıştır.

3.1. Çözüm Uzayı (Solution Space)

Tabu arama gibi modern sezgisel yöntemlerin başarısını etkileyen en kritik kararlardan biri çözümün nasıl temsil edileceğinin belirlenmesidir. Bu çalışmada, daha önce geliştirilmiş tavlama benzetimi ve tabu arama algoritmalarında olduğu gibi hangi işin hangi istasyona atandığını gösteren bir çözüm temsilinin aksine, genellikle evrimsel algoritmalarda kullanılan göreve yönelik bir temsil kullanılmıştır. Göreve yönelik çözüm temsilinde, görevler istasyonlara atanma sıralarına göre birbiri ardına listelenirler. Eğer görev sırası öncelik kısıtlarını ihlal etmiyorsa çözümün olurlu olduğu söylenir. Örneğin, Şekil 1'deki öncelik diyagramı dikkate alındığında, <1,2,6,3,5,4,7,8,9,10,11> görev sırasının oluşturduğu çözüm olurludur. Geliştirilen tabu arama algoritmasının araştırdığı çözüm uzayı sadece olurlu çözümleri içermektedir.

3.2. Çözümleme (Decoding)

Kullanılan çözüm temsili, hangi görevin hangi istasyona atandığı hakkında bir ipucu vermediği için tip-2 dengeleme probleminin amaç fonksiyonu olan maksimum istasyon zamanının doğrudan hesaplanması mümkün değildir. İstasyon sayısı (m) bilindiğinde çevrim süresi, aşağıda adımları verilen, Kim vd. [13]'nin geliştirdiği yöntemle belirlenmektedir:

Adım 1: Teorik minimum çevrim süresine eşit başlangıç çevrim süresini (BT) hesapla (BT= T/m)

Adım 2: (m-1).inci istasyona kadar çevrim süresini BT olarak görevleri istasyonlara ata. Kalan tüm görevleri son istasyon m'e ata.

Adım 3: W_j , (j=1, 2, ...,m) ve W_j^+ , (j=1, 2, ..., m-1) değerlerini hesapla. Burada W_j , j istasyonunun iş yükünü, W_j^+ ise W_j ile (j+1).inci istasyona atanmış ilk görevin zaman toplamından oluşan j istasyonunun potansiyel iş yükünü göstermektedir.

Adım 4: $DT = \max \{W_j \mid j=1, 2, \dots, m\}$ ve $BT = \min \{W_j^+ \mid j=1, 2, \dots, m-1\}$ değerlerini hesapla. Eğer $DT \leq BT$ ise dur, DT minimum çevrim süresidir. Aksi halde, adım 2'ye git.

3.3. Başlangıç Çözümü (Initial Solution)

Başlangıç çözümü için herhangi bir sezgisel ya da çözümün olurluluğunu garanti eden rassal bir prosedür kullanılabilir. Olurlu bir çözümü rassal olarak belirleyen bir yöntem aşağıda verilmiştir [13]:

Adım 1: Öncülü olmayan işlerden bir başlangıç listesi oluştur ve boş bir dizi yarat.

Adım 2: Mevcut liste boş ise dur. Aksi halde, adım 3'e git.

Adım 3: Mevcut listeden rassal olarak bir görev seç ve diziyeye ekle.

Adım 4: Mevcut listeyi, seçilen görevi silerek ve görevin doğrudan ardılarını, eğer bu ardılların öncülleri zaten dizide iseler listeye ekleyerek, güncelle. Adım 2'ye git.

3.4. Amaç Fonksiyonu (Objective Function)

Matematiksel model, çevrim süresinin minimizasyonu ve istasyon iş yüklerinin ortalama iş yükünden sapmalarının toplamının minimizasyonu olmak üzere iki amaca sahiptir. Problemin çok amaçlı yapısını temsil etmek için amaçlar ilk önce oransal hale getirilmiş, sonra da Eş. 11'deki gibi bütünlük bir fonksiyon olarak ifade edilmiştir.

$$\text{Min } \alpha \cdot \left(\frac{CT}{CT_{\min}} \right) + \beta \cdot \left(\frac{\sum_{j=1}^m (U_j + O_j)}{T} \right) \quad (11)$$

Bütünlük amaç fonksiyonunun ilk bölümü çevrim süresi amacına, ikinci bölümü ise iş yükü dengesizliği amacına karşılık gelmektedir. α ve β , amaçların ağırlıklarını temsil etmektedir. Çevrim süresi amacı öncelikli olduğu için algoritma $\alpha=200$, $\beta=100$ ağırlıklarıyla çalıştırılmıştır.

3.5. Hareket Mekanizması (Move Mechanism)

Mevcut çözümden komşu bir çözüme geçişte araya girme ve yer değiştirme hareketlerinden yararlanılmıştır. İlk önce, hangi hareketin gerçekleştirileceğine rassal olarak karar verilmektedir. Her iki hareketin de yapılma olasılıkları eşit, yani 0,50 olarak alınmıştır. Yapılacak hareket seçildikten sonra hareketin gerçekleştirileceği çözüm parçası belirlenmektedir. Hareket, 0,50 olasılıkla tüm çözüm içindeki görevlerden biri seçilerek, 0,50 olasılıkla da en yüklü istasyondaki görevler dikkate alınarak gerçekleştirilmektedir. Aslında, mevcut çözüm, hareket, ancak, en yüklü istasyondan gerçekleştirilirse iyileşme ihtimaline sahiptir. Yapılacak bazı hareketlerin en yüklü istasyon haricindeki istasyonlara atanan görevleri de içermesi aramaya çeşitlilik kazandıracaktır. Bölüm 3.2'de verilen çözümleme metodu kullanılarak mevcut çözümün çevrim süresinin belirlenmesi ve bu çevrim süresine göre görevlerin istasyonlara atanmasıyla, göreve dayalı her bir çözümdeki en yüklü istasyon ve bu istasyona atanan görevleri belirli hale getirmek mümkündür. Hareketin kaynağı olan her görev için, daha önce belirlenmiş harekete göre, çözümün olurluluğu bozulmadan görevin araya girebileceği pozisyonlar veya yer değiştirebileceği diğer görevler bir diziyeye aktarılır, en çok iyileşen stratejisi yardımıyla, yani dizideki tüm alternatifler incelenerek, mevcut çözüm güncellenir. Eğer hareketin gerçekleştirileceği görev tüm çözümden seçilecekse, rassal olarak sadece bir görev belirlenerek görevin komşuluğu araştırılır. Hareketin gerçekleştirileceği çözüm parçası

maksimum yüke sahip istasyona atanan görevleri içeriyorsa, bu istasyondaki her bir görevin komşuluğu araştırılarak mevcut çözüm güncellenir.

3.6. Tabu Listeleri ve Tabu Süreleri (Tabu Lists and Tabu Tenures)

Yapılacak bir hareketin tabu olup olmadığının kontrolü için $(n \times n)$ boyutlu, ilgili hareketin tabu şartının sona ereceği iterasyonu kaydeden bir tabu listesi kullanılmıştır. Eğer herhangi bir iterasyonda i görevi k_1 sırasından k_2 sırasına taşındıysa, i görevinin k_1 sırasına geri dönmesi tabu süresi (tt) kadar iterasyon boyunca engellenmektedir. Bu amaçla, hareketin tabu şartının bitişi Eş. 12 ile güncellenmektedir.

$$\text{tabuend}[k_1, i] = \text{güncel iterasyon} + tt \quad (12)$$

Yapılan harekete göre, Eş. 13 veya Eş. 14 şartları sağlanıyorsa ilgili hareket tabu olarak adlandırılır:

Araya girme hareketi: k_1 sırasındaki i görevi k_2 sırasına taşınıyorsa,

$$\text{güncel iterasyon} \leq \text{tabuend}[k_2, i] \quad (13)$$

Yer değiştirme hareketi: k_1 sırasındaki i görevi ile k_2 sırasındaki j görevi yer değiştiriyorsa,

$$\begin{aligned} \text{güncel iterasyon} &\leq \text{tabuend}[k_1, j] \wedge \\ \text{güncel iterasyon} &\leq \text{tabuend}[k_2, i] \end{aligned} \quad (14)$$

Tabu süreleri statik olarak alınmış ve farklı problem büyüklüklerine orantılı olarak görev sayısının karekökünün en yakın tamsayıya yuvarlanmasıyla hesaplanmıştır (Eş. 15).

$$tt = \text{round}(\sqrt{n}) \quad (15)$$

3.7. Tabu Yıkma Kriteri (Aspiration Criterion)

Bu çalışmada, amaca göre tabu yıkma kullanılmıştır. Buna göre tabu olan bir hareket, mevcut iterasyona kadar bulunmuş en iyi çözümden daha iyi bir çözümün elde edilmesini sağlıyorsa, tabu olmasına rağmen gerçekleştirilmektedir. Bununla birlikte, eğer mevcut çözümden olurluluk bozulmadan yapılabilecek tüm hareketler tabu ise mevcut çözüm, tabu durumu bitmeye en yakın hareket ile güncellenmektedir.

3.8. Çeşitlendirme Stratejisi (Diversification Strategy)

Aramanın çeşitlendirilmesini sağlamak amacıyla hareket değerlerinin değiştirilmesi stratejisi kullanılmıştır. Hareket değerlerinin değiştirilmesinde, uzun dönem hafızada tutulan "kalıcılık sıklığı" bilgisinden yararlanılmıştır. Kalıcılık sıklığı ile mevcut çözümlerde i görevinin kaç kez k sırasını işgal ettiğinin kaydı tutulur. Bu sayede, arama boyunca farklı görevlerin daha çok işgal ettiği sıralar belirlenir ve belli görevlerin daha az bulunduğu sıralara yol açacak hareketler teşvik edilerek aramanın yönü değiştirilebilir. Kalıcılık

sıklığı, mevcut çözümde hangi görevin hangi sırada olduğu bilgisi dikkate alınarak her iterasyonda güncellenir. Örneğin, ilgili iterasyonun mevcut çözümünde i görevi k sırasında bulunuyorsa, kalıcılık sıklığı Eş. 16'daki gibi değiştirilir.

$$\text{frequency}[i, k] = \text{frequency}[i, k] + 1 \quad (16)$$

Hareket değerlerinin değiştirilmesi, kalıcılık sıklığının ceza bilgilerine dönüştürülmesi vasıtasıyla gerçekleştirilir. Her iterasyonda, yeni mevcut çözüme geçerken değiştirilen değişken ya da değişkenlere ait kalıcılık sıklığı oranı bir ceza katsayısı (penalty) ile çarpılarak orjinal amaç fonksiyonuna eklenir ve değiştirilmiş bir amaç fonksiyonu değeri elde edilir. Yapılan hareketin cinsine göre, Eş. 17 ve Eş. 18'de verilmiş olan değiştirilmiş amaç fonksiyonu (modified $f(x)$), mevcut çözümün komşuları arasından en iyisinin belirlenmesinde kullanılır.

Araya girme hareketinin yapıldığı durum için: Eğer i görevi k sırasına taşınıyorsa;

$$\text{modified}_f(x) = f(x) + \text{penalty} * \left(\frac{\text{frequency}[i, k]}{\text{güncel iterasyon}} \right) \quad (17)$$

- Yer değiştirme hareketinin yapıldığı durum için: Eğer k_1 sırasındaki i görevi ile k_2 sırasındaki j görevi yer değiştiriyorsa;

$$\begin{aligned} \text{modified}_f(x) &= f(x) + \text{penalty} * \\ &\left(\frac{\text{frequency}[i, k_2] + \text{frequency}[j, k_1]}{2 * (\text{güncel iterasyon})} \right) \end{aligned} \quad (18)$$

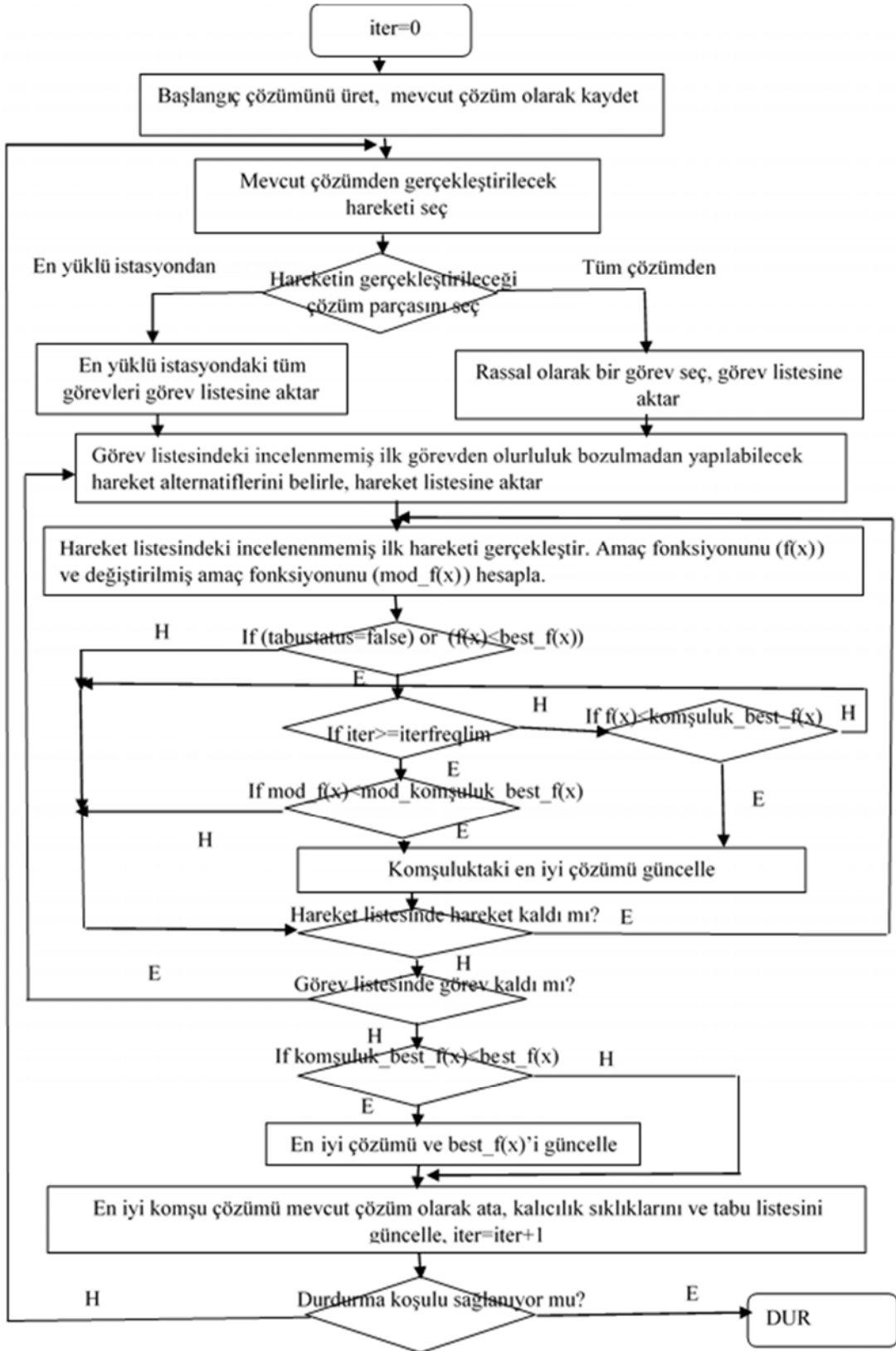
Algoritma sırasında kalıcılık sıklıkları ilk iterasyondan itibaren tutulurken, değiştirilmiş amaç fonksiyonuna göre seçim belli bir iterasyon limitinden (iterfreqlim) sonra yapılmaktadır. Bu çalışmada, söz konusu limit 1000 olarak belirlenmiştir. Algoritmada kullanılacak ceza katsayısı ise 0,025 olarak seçilmiştir.

3.9. Durdurma Koşulu (Stopping Criterion)

Algoritma, en iyi çözüm değişmeden belli bir iterasyon sayısı geçtikten sonra durdurulmaktadır. Bu sayı, problem büyüklüğüne bağlıdır ve $n * 200$ olarak belirlenmiştir. TurboPascal programlama dili ile kodlanan algoritmanın akış şeması Şekil 2'de verilmiştir.

4. HESAPLAMA SONUÇLARI (COMPUTATIONAL RESULTS)

Geliştirilen tabu arama algoritmasının iki amaçlı tip-2 basit montaj hattı dengeleme problemlerinin çözümündeki etkinliği açık literatürden alınmış 9 probleme ait 23 örnek üzerinde test edilmiştir. Algoritma TurboPascal dilinde kodlanmış ve tüm denemeler 2,00 GB ön belleğe sahip Pentium çift çekirdekli 2,60 Ghz işlemcili kişisel bir bilgisayar üzerinde gerçekleştirilmiştir. Algoritma, önceki bölümde verilen parametre değerleri ile her problem için değişik rassal sayılarla onar defa çalıştırılmıştır.



Şekil 2. Tabu arama algoritmasının akış şeması (Flow chart of the tabu search algorithm)

Tablo 2. Geliştirilen TA algoritması ve Uğurdağ vd. [7]'nin yönteminden elde edilen çevrim süresi sonuçlarının ve optimal çözüm değerlerinin karşılaştırılması

(Comparison of the results obtained from the developed tabu search algorithm, Ugurdağ etc. [7]'s method and the optimal solution values)

Problem	İst. sayısı	Uğurdağ'ın yöntemi			Geliştirilen TA algoritması			
		Optimum çevrim süresi	Çevrim süresi	Optimum dan % sapma	En iyi Çevrim süresi	Ort. Çevrim süresi	En kötü Çevrim süresi	Optimum dan % sapma
Merten-7	5	7	7	0,00	7	7	7	0,00
	3	10	11	10,00	10	10	10	0,00
	2	15	15	0,00	15	15	15	0,00
Jaeschke-9	7	7	7	0,00	7	7	7	0,00
	4	10	10	0,00	10	10	10	0,00
	3	13	14	7,69	13	13	13	0,00
Jackson-11	5	10	11	10,00	10	10	10	0,00
	4	12	13	8,33	12	12	12	0,00
	3	16	16	0,00	16	16	16	0,00
Mitchell-21	8	14	14	0,00	14	14	14	0,00
	5	21	23	9,52	21	21	21	0,00
	3	35	36	2,86	35	35	35	0,00
Heskiaoff-28	5	205	210	2,44	205	205	205	0,00
	4	256	258	0,78	256	256	256	0,00
Sawyer-30	13	26	28	7,69	26	26	26	0,00
	8	41	43	4,88	41	41	41	0,00
	5	65	66	1,54	65	65	65	0,00
Killbridge and Wester-45	10	56	57	1,79	56	56	56	0,00
	6	92	93	1,09	92	92	92	0,00
	3	184	185	0,54	184	184	184	0,00
Tonge-70	11	320	325	1,56	320	320	320	0,00
	8	439	445	1,37	439	439,2	440	0,05
Arcus-111	9	16711	16796	0,51	16712	16714	16716	0,02
Optimalden ortalama % sapma				3,156				0,003

Algoritmanın performansı, ulaşılan sonuçların, çevrim süresi amacı için aynı problem setini kullanan Uğurdağ vd. [7]'nin sonuçları ve problemlerin optimum çözüm değerleri ile, iş yükü dengeleme amacı için ise, ikinci bölümde anlatıldığı gibi oluşturulan dengesizliğin minimizasyonu amaçlı matematiksel modellerin GAMS çözümlerinden elde edilen sonuçlarla karşılaştırmasıyla ölçülmüştür. Algoritmanın çevrim süresi ve iş yükü dengeleme amaçları için performansı ayrı ayrı Tablo 2 ve Tablo 3'te görülmektedir.

Tablo 2'de ilk iki sütunda, sırasıyla, problem isimleri ve istasyon sayıları verilmiştir. Problem isimlerinin yanındaki sayılar, problemin sahip olduğu görev sayısını göstermektedir. Geliştirilen TA algoritması, 21 örnekte her denemede, 8 istasyonlu Tonge probleminde 10 denemenin 8'inde optimal çevrim süresine ulaşmış, 9 istasyonlu Arcus probleminde ise hiç bir denemede optimal sonucu elde edememiştir. Buna rağmen, söz konusu problemde elde edilen ortalama çevrim süresi değerinin optimalden sapması %0,02'dir. Bununla birlikte, Uğurdağ'ın yöntemi sadece 6 problemde optima ulaşırken geliştirilen tabu arama algoritması 23 problemin 22'sinde optimal çevrim sürelerini elde etmiştir. Uğurdağ'ın yöntemi için optimumdan ortalama sapma %3,156 iken geliştirilen TA algoritması için %0,003'tür. Ayrıca, tüm örneklerde TA algoritmasının

bulduğu en kötü çözüm değeri en az Uğurdağ'ın yöntemi ile ulaşılan sonuç kadar iyidir. Tablo 3'te ise çevrim süresi ve iş yükü dengesizliği için çözdürülen GAMS modellerinin sonuçları ile TA algoritmasından elde edilen sonuçlar karşılaştırılmış, algoritmanın ortalama koşum zamanları verilmiştir. Her bir problem için oluşturulan GAMS modelleri ilk önce çevrim süresi amacı, sonra da iş yükü dengesizliği amacıyla çalıştırılmıştır. Büyük boyutlu problemlerin optimal sonuca ulaşmasının uzun sürebileceği ihtimaline karşın 7200 sn'lik bir süre sınırı tanımlanmıştır.

Tabloda çevrim süresi amaç değerlerinin tekrar verilmesinin nedeni, amaçlar arasındaki öncelik ilişkisinden dolayı, iş yükü dengesizliği amacının iyiliğinin çevrim süresi amacıyla birlikte değerlendirildiğinde ortaya çıkmasıdır. Parantez içindeki değerlerin ilki çevrim süresi, ikincisi iş yükü dengesizliği amacını göstermektedir. Tablo 3'ün 4 ve 7.inci sütunlarında ilgili yöntemle elde edilen sonucun iş yükü dengesizliği oranları verilmiştir. Bu oran toplam dengesizliğin toplam iş yüküne oranlanması ile elde edilmektedir. 8.inci sütunda TA algoritması ve GAMS sonuçları arasındaki oransal dengesizlik farkları, son sütunda ise algoritmanın ortalama koşum zamanları görülmektedir. Killbridge ve Wester'in 10 istasyonlu, Tonge'nin 11 ve 8 istasyonlu iş yükü dengesizliği amaçlı problemleri, Arcus'un

Tablo 3. Çevrim süresi ve iş yükü dengesizliği amaç fonksiyon değerleri için GAMS ve TA sonuçlarının karşılaştırılması
(Comparison of the cycle time and work load imbalance objective functions obtained from GAMS and TS algorithm)

Problem	İst. Say.	GAMS çözümü	Geliştirilen TA algoritması					
			GAMS çözümü deng. oranı (1)	En iyi	Ortalama	Deng. oranı (2)	İş yükü deng. farkı (2)-(1)	Ort. koşum zamanı (CPU sn)
Merten-7	5	(7; 3,20)	0,1103	(7; 3,20)	(7; 3,20)	0,1103	0,000	0,72
	3	(10; 1,33)	0,0460	(10; 1,33)	(10; 1,33)	0,0460	0,000	0,70
	2	(15; 1,00)	0,0345	(15; 1,00)	(15; 1,00)	0,0345	0,000	0,69
Jaeschke-9	7	(7; 4,86)	0,1313	(7; 4,86)	(7; 4,86)	0,1313	0,000	0,46
	4	(10; 1,50)	0,0405	(10; 1,50)	(10; 1,50)	0,0405	0,000	0,46
Jackson-11	3	(13; 1,33)	0,0360	(13; 1,33)	(13; 1,33)	0,0360	0,000	0,46
	5	(10; 3,20)	0,0696	(10; 3,20)	(10; 3,20)	0,0696	0,000	0,66
Mitchell-21	4	(12; 2,00)	0,0435	(12; 2,00)	(12; 2,00)	0,0435	0,000	0,59
	3	(16; 1,33)	0,0290	(16; 1,33)	(16; 1,33)	0,0290	0,000	0,62
Heskiaoff-28	8	(14; 3,50)	0,0333	(14; 3,50)	(14; 3,50)	0,0333	0,000	1,49
	5	(21; 0,00)	0,0000	(21; 0,00)	(21; 0,00)	0,0000	0,000	1,37
	3	(35; 0,00)	0,0000	(35; 0,00)	(35; 0,00)	0,0000	0,000	1,51
Sawyer-30	5	(205; 1,60)	0,0016	(205; 1,60)	(205; 1,60)	0,0016	0,000	4,89
	4	(256; 0,00)	0,0000	(256; 0,00)	(256; 0,00)	0,0000	0,000	4,88
Killbridge and	13	(26; 5,54)	0,0171	(26; 5,54)	(26; 5,54)	0,0171	0,000	5,55
	8	(41; 4,00)	0,0123	(41; 4,00)	(41; 4,00)	0,0123	0,000	4,44
	5	(65; 1,60)	0,0049	(65; 1,60)	(65; 1,60)	0,0049	0,000	3,20
Wester-45	10	(56; 3,20 ^b)	0,0058	(56; 3,20)	(56; 3,20)	0,0058	0,000	13,43
	6	(92; 0,00)	0,0000	(92; 0,00)	(92; 0,00)	0,0000	0,000	11,86
Tonge-70	3	(184; 0,00)	0,0000	(184; 0,00)	(184; 0,00)	0,0000	0,000	12,63
	11	(320; 1,82 ^a)	0,0005	(320; 1,82)	(320; 5,27)	0,0015	0,0001	77,00
Arcus-111	8	(439; 3,00 ^a)	0,0009	(439; 3,0)	(439,2; 3,3)	0,00094	0,00004	67,93
	9	(16714 ^a ; 12 ^a)	0,00008	(16712; 10)	(16714; 16,4)	0,0001	0,00002	924,96

^aVerilen zaman sınırı (7200 sn) içerisinde tam sayı çözüm elde edilmiştir.

^bHafıza yetersizliğinden dolayı, GAMS çözümü 3095 sn. içinde sonlandırılmıştır.

9 istasyonlu hem çevrim süresi hem de iş yükü dengesizliği amaçlı problemleri hariç tüm test örneklerinde, 7200 saniyelik süre sınırı içerisinde, GAMS programının CPLEX çözücü ile optimal sonuçlara ulaşılmıştır. Geliştirilen TA algoritmasının, GAMS ile optimal bulunan tüm problemlerde iş yükü dengeleme amacı için her denemede optimal sonucu elde ettiği görülmektedir. GAMS ile optima ulaşılmayan problemlerde, 10 istasyonlu Killbridge-Wester örneği için her denemede GAMS'in iş yükü dengesizliği sonucu yakalanırken, 11 ve 8 istasyonlu Tonge ve 9 istasyonlu Arcus problemleri için ortalama, GAMS'in 7200 sn.'de ulaştığından, daha kötü dengesizlik değerleri elde edilmiştir. Buna rağmen, oransal olarak değerlendirildiğinde, söz konusu test örnekleri için TA algoritmasının ortalama iş yükü dengesizliği değerlerinin GAMS çözümlerinden farkı, sırasıyla, %0,01, %0,004 ve %0,002'dir. Ayrıca, algoritmanın küçük ve orta boyutlu problemlerde hızlı bir şekilde çözüme ulaştığı, 9 istasyonlu Arcus probleminde ise ortalama yaklaşık 15 dakikalık bir süreye ihtiyaç duyduğu görülmektedir. Ancak, bu süre, özellikle büyük boyutlu problemlerin GAMS koşum zamanlarından çok daha düşüktür. Bununla birlikte, matematiksel modelin iki aşamada çözüldüğü dikkate alındığında, GAMS programıyla hızlı bir şekilde optimum

çözüme ulaşılan küçük boyutlu problemler için de geliştirilen tabu algoritmasının daha pratik olduğu açıktır.

5. SONUÇLAR (CONCLUSIONS)

Birçok farklı çeşide sahip olan montaj hattı dengeleme problemleri NP-zor problem sınıfına girdiğinden kesin yöntemlerle optimal çözüme ulaşmak oldukça zor ve zaman alıcıdır [31]. Bu nedenle, literatürde genellikle meta-sezgisel tekniklerle ele alınmışlardır. Bu çalışmada ise istasyonlar arası iş yükü dengesizliğini azaltmayı ikinci amaç olarak dikkate alan tip-2 basit montaj hattı dengeleme probleminin çözümleri için bir tabu arama algoritması önerilmiştir. Önerilen algoritmanın literatürde aynı problem için geliştirilmiş diğer tabu arama algoritmalarından en önemli farkı, görevlerin istasyonlara atanma sırasına göre listelendiği göreve yönelik bir çözüm temsili kullanması ve iki amacı dikkate almasıdır. Algoritmanın etkinliği açık literatürden alınmış 9 probleme ait 23 örnek üzerinde test edilmiştir. Yapılan denemeler sonucunda, algoritmanın çevrim süresi amacı için 23 örneğin 22'sinde, iş yükü dengesizliği amacı için de 23 örneğin 20'sinde hemen her koşulda optimal sonuçlara ulaştığı, diğer örnekler için de optima çok yakın sonuçlar elde ettiği görülmektedir.

Algoritmayı basit montaj hattı dengeleme problemlerinin diğer çeşitlerine ve farklı tipteki montaj hatlarına uyarlayarak çalışmayı genişletmek mümkündür. Ayrıca, geliştirilen algoritmanın farklı ve daha büyük problemler içeren problem setleri üzerinde denenmesi ve farklı çözüm temsillerinin tabu arama ya da benzer arama algoritmalarının performansını üzerindeki etkisinin araştırılması da düşünülmektedir.

6. SİMGELER (SYMBOLS)

CT	çevrim süresi
CT_{max}	çevrim süresi için üst sınır değeri
CT_{min}	çevrim süresi için alt sınır değeri
$E_i(CT_{max})$	CT_{max} değerine göre i görevinin atanabileceği en erken istasyon (Eş. 19)
	$(E_i(CT_{max})) = \left\lceil \frac{t_i + \sum_{k \in PT_i} t_k}{CT_{max}} \right\rceil \quad (19), [32]$
FS_i	i görevinin atanabileceği istasyonlar kümesi (her görev için $[E_i, L_i]$ hesaplamaları sonucu belirlenir)
FT_j	j istasyonuna atanabilecek görevler kümesi (her görev için $[E_i, L_i]$ hesaplamaları sonucu belirlenir)
$L_i(CT_{max})$	CT_{max} değerine göre i görevinin atanabileceği en geç istasyon (Eş. 20)
	$(L_i(CT_{max})) = m + 1 - \left\lfloor \frac{t_i + \sum_{k \in ST_i} t_k}{CT_{max}} \right\rfloor \quad (20), [32]$
m	istasyon sayısı ($j=1, \dots, m$)
n	görev sayısı ($i=1, \dots, n$)
O_j	j istasyonu için ortalama iş yükünden yukarı sapma
PR	aralarında doğrudan öncelik ilişkisi bulunan görev çiftleri kümesi
PT_i	i görevinden önce gelen görevler kümesi
ST_i	i görevinden sonra gelen görevler kümesi
t_i	i görevinin işlem zamanı
T	görev zamanları toplamı
U_j	j istasyonu için ortalama iş yükünden aşağı sapma
x_{ij}	Eğer i görevi j istasyonuna atanırsa, 1; aksi halde, 0

KAYNAKLAR (REFERENCES)

- Scholl A., Voß S., Simple Assembly Line Balancing – Heuristic Approaches, Journal of Heuristics, 2, 217-244, 1996.
- Rachamadugu R. ve Talbot B., Improving the equality of workload assignments in assembly lines, Int. J. Prod. Res., 29 (3), 619-633, 1991.
- Mastor A.A., An experimental investigation and comparative evaluation of production line balancing techniques, Manage. Sci., 16 (11), 728-746, 1970.
- Gehrline W.V. ve Patterson J.H., Sequencing for assembly lines with integer task times, Manage. Sci., 21 (9), 1064-1070, 1975.
- Hackman S.T., Magazine M.J., Wee T.S., Fast, effective algorithms for simple assembly line balancing problems, Oper. Res., 37, 916-924, 1989.
- Klein R. ve Scholl A., Maximizing the production rate in simple assembly line balancing – A branch and bound procedure, Eur. J. Oper. Res., 91, 367-385, 1996.
- Uğurdağ H.F., Rachamadugu R., Papachristou C.A., Designing paced assembly lines with fixed number of stations, Eur. J. Oper. Res., 102, 488-501, 1997.
- Liu S.B., Ong H.L., Huang H.C., Two bi-directional heuristics for the assembly line type II problem, Int. J. Adv. Manuf. Technol., 22, 656-661, 2003.
- Kılınççı Ö., A petri-net based heuristic for simple assembly line balancing problem of type 2, Int. J. Adv. Manuf. Technol., 46, 329-338, 2010.
- Blum C., Iterative beam search for simple assembly line balancing with a fixed number of work stations, Statistics and Operations Research Transactions, 35(2), 145-164, 2011.
- La Scalia G., Rosa M., Giuseppe A., Mario E., Solving type-2 assembly line balancing problem with fuzzy binary linear programming, J. Intell. Fuzzy Syst., 25, 517-524, 2013.
- Heinrici A., A comparison between simulated annealing and tabu search with an example from the production planning, Operations Research Proceedings 1994, Editör: Dyckhoff H., Derigs U., Salomon M. ve Tijms H. C., Springer Verlag, Berlin, 498-503, 1994.
- Kim Y.K., Kim Y.J., Kim Y., Genetic algorithms for assembly line balancing with various objectives, Comput. Ind. Eng., 30 (3), 397-409, 1996.
- Nearchou A.C., Balancing large assembly lines by a heuristic based on differential evolution method, Int. J. Adv. Manuf. Technol., 34, 1016-1029, 2007.
- Nearchou A.C., Multi-objective balancing of assembly line by population heuristics, Int. J. Prod. Res., 46 (8), 2275-2297, 2008.
- Nearchou A.C., Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization, Int. J. Prod. Econ., 129, 242-250, 2011.
- Zacharia P.Th. ve Nearchou A.C., Multi-objective fuzzy assembly line balancing using genetic algorithms, J. Intell. Manuf., 23, 615-627, 2012.
- Zheng Q., Li M., Li Y., Tang Q., Station ant colony optimization for the type 2 assembly line balancing problem, Int. J. Adv. Manuf. Technol., 66, 1859-1870, 2013.
- Mozdgir A., Mahdavi I., Badeleh I.S., Solimanpur M., Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing, Math. Comput. Modell., 57, 137-151, 2013.
- Zacharia P.T., Tsirkas S.A., Kabouridis G., Giannopoulos G. I., Planning the construction process of a robotic arm using a genetic algorithm, Int. J. Adv. Manuf. Technol., 79, 1293-1302, 2015.
- Zhang H., Yan Q., Liu Y., Jiang Z., An integer-coded differential evolution algorithm for simple assembly line balancing of type 2, Assembly Autom., 36(3), 246-261, 2016.
- Polat O., Kalaycı C.B., Mutlu Ö., Gupta S.M., A two-phase variable neighbourhood search algorithm for

- assembly line worker assignment and balancing problem type-II: an industrial case study, *Int. J. Prod. Res.*, 54 (3), 722-74, 2016.
23. Lei D. ve Guo X., Variable neighborhood search for the second type of two-sided assembly line balancing problem, *Comput. Oper. Res.*, 72, 183–188, 2016.
 24. Rabbani M., Mousavi Z., Asl H.F., Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem, *J. Ind. Prod. Eng.*, 33 (7), 472-484, 2016.
 25. Roshani A. ve Giglio D., Simulated annealing algorithms for the multi-manned assembly line balancing problem: minimising cycle time, *Int. J. Prod. Res.*, 55 (10), 2731-2751, 2017.
 26. Şahin M. ve Kellegöz T., An efficient grouping genetic algorithm for U-shaped assembly line balancing problems with maximizing production rate, *Memetic Comp.*, 9, 213–229, 2017.
 27. Driscoll J. ve Thilakawardana D., The definition of assembly line balancing difficulty and evaluation of balance solution quality, *Rob. Comput. Integr. Manuf.*, 17, 81-86, 2001.
 28. Scholl A., *Data of assembly line balancing problems*, Schriften zur Quantitativen Betriebswirtschaftslehre 16/93, TH Darmstadt, 1993.
 29. Glover F., Tabu Search-Part I, *ORSA Journal on Computing*, 1 (3), 190-206, 1989.
 30. Glover F. ve Laguna M., *Tabu Search*, Kluwer Academic Publishers, Boston, 1997.
 31. Yağmahan B. ve Emel E., Mixed-model assembly line balancing with smoothing approach based on tabu search algorithm, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 30 (1), 57-69, 2015.
 32. Pastor R. ve Ferrer L., An improved mathematical program to solve simple assembly line balancing problem, *Int. J. Prod. Res.*, 47 (11), 2943-2959, 2009.

