



An approach for solving single-machine bi-criteria scheduling problem

Banu Lokman*^{ID}

University of Portsmouth, Portsmouth Business School, Centre of Operations Research and Logistics, PO1 3DE, Portsmouth, United Kingdom

Highlights:

- A single machine scheduling problem that minimizes the maximum weighted tardiness and total weighted flow time
- Design of an algorithm that is capable of generating all nondominated points as well as a representative set with a desired quality level
- Extensive performance analysis on test problems

Keywords:

- Bi-criteria single machine scheduling
- Nondominated point
- Representative set
- Maximum weighted tardiness
- Total weighted flow time

Article Info:

Research Article
Received: 04.06.2019
Accepted: 12.05.2020

DOI:

10.17341/gazimmfd.573734

Correspondence:

Author: Banu Lokman
e-mail:
banu.lokman@port.ac.uk
phone: +44 746 401 0779

Graphical/Tabular Abstract

Most real-world scheduling problems involve multi-objective optimization since the decision makers consider multiple criteria that compete against each other. Since these problems do not usually have a unique optimal solution, one of the most common approaches is to generate the whole or partial set of nondominated points with a desired quality level. These points correspond to the efficient schedules for which it is not possible to improve any criterion without worsening some other. In this study, we develop an iterative approach that generates efficient schedules of a bi-criteria scheduling problem that considers the maximum weighted tardiness and total weighted flow time.

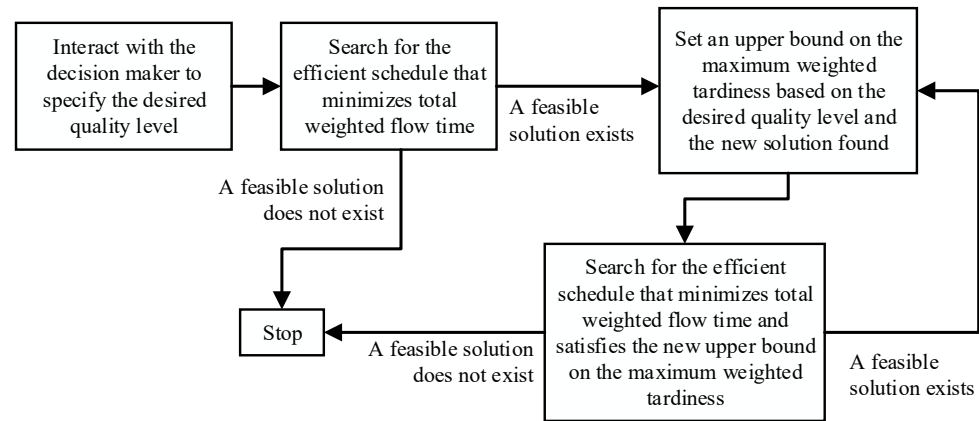


Figure A. The proposed approach for bi-criteria scheduling problem on a single machine

Purpose: The aim of this study is to solve a bi-criteria scheduling problem that minimizes the maximum weighted tardiness and total weighted flow time.

Theory and Methods:

A multi-objective approach is developed to find efficient schedules that consider the maximum weighted tardiness and total weighted flow time. It utilizes an epsilon-constraint method that imposes an upper bound on the maximum weighted tardiness value and finds the efficient schedule with minimum total weighted flow time at each iteration. Each solution is found by using a modification of Smith (1956)[10]'s backward scheduling rule. By systematically changing the upper bound on the maximum weighted tardiness, the algorithm generates a set of nondominated points that guarantee to represent the whole nondominated set based on a prespecified quality measure.

Results:

The results of computational experiments show that the number of nondominated points is increasing substantially with the problem size since there is a tradeoff between the two criteria: maximum weighted tardiness and total weighted flow time. Therefore, finding a representative set that meets the desired quality level is an effective solution method in terms of the solution time and the number of schedules presented to the decision maker.

Conclusion:

Instead of generating all efficient schedules, finding a subset that represents the entire set based on a certain quality measure is a practical approach in real-world scheduling problems.



İki kriterli tek makinalı çizelgeleme problemi için bir yaklaşım

Banu Lokman*^{ID}

University of Portsmouth, Portsmouth Business School, Centre of Operations Research and Logistics, PO1 3DE, Portsmouth, United Kingdom

Ö N E Ç İ K A N L A R

- Maksimum ağırlıklı geç bitirme ve ağırlıklı toplam akış zamanının en küçüklendiği tek makinalı çizelgeleme problemi
- Hem tüm baskın noktaları hem de istenilen kalite seviyesinde temsili bir altküme üretebilen algoritma tasarımı
- Test problemleri üzerinde kapsamlı performans analizi

Makale Bilgileri

Araştırma Makalesi

Geliş: 04.06.2019

Kabul: 12.05.2020

DOI:

10.17341/gazimmfd.573734

Anahtar Kelimeler:

İki kriterli tek makinalı
çizelgeleme,
baskın nokta,
temsili küme,
maksimum ağırlıklı geç
bitirme,
ağırlıklı toplam akış zamanı

ÖZET

Bu çalışmada, maksimum ağırlıklı geç bitirme ve ağırlıklı toplam akış zamanının en küçüklendiği iki kriterli tek makinalı çizelgeleme problemi için bir yaklaşım önerilmiştir. Tüm baskın noktaları üretebilen bu algoritma; aynı zamanda karar vericinin belirlediği bir kalite seviyesine göre baskın nokta kümesini temsil edebilen bir altküme bulmayı da garantilemektedir. Geliştirilen yaklaşım, farklı büyüklükteki ve farklı özelliklerdeki problemler üzerinde farklı kalite seviyelerinde uygulanmıştır. Deney sonuçları, algoritmamızın hem çözüm süresi hem karar vericiye sunulan çizelge sayısı açısından iyi bir yaklaşım olduğunu göstermektedir.

An approach for solving single-machine bi-criteria scheduling problem

H I G H L I G H T S

- A single machine scheduling problem that minimizes the maximum weighted tardiness and total weighted flow time
- Design of an algorithm that is capable of generating all nondominated points as well as a representative set with a desired quality level
- Extensive performance analysis on test problems

Article Info

Research Article

Received: 04.06.2019

Accepted: 12.05.2020

DOI:

10.17341/gazimmfd.573734

Keywords:

Bi-criteria single machine
scheduling,
nondominated point,
representative set,
maximum weighted
tardiness,
total weighted flow time

ABSTRACT

In this study, we develop an algorithm for a bi-criteria scheduling problem that minimizes the maximum weighted tardiness and total weighted flow time. While the algorithm is capable of generating all nondominated points, the algorithm is designed to generate a subset of the nondominated points that guarantee to represent the whole nondominated set based on a prespecified quality measure. We apply the algorithm on a wide variety of problems with different properties. The results of the experiments show that the algorithm works well not only in terms of the solution time but also the number of schedules presented to the decision maker.

*Sorumlu Yazar/Corresponding Author: *banu.lokman@port.ac.uk/ Tel: +44 746 401 0779

1. GİRİŞ (INTRODUCTION)

Günlük hayat çizelgeleme problemlerinde karar vericiler çoğu zaman birbiri ile çelişen birden fazla kriter ile karşı karşıyadır. Bu problemlerde, her kriterde iyi olan tek bir çözüm yoktur ve amaç etkin çözümleri bulmaktır. Etkin çözümler bir kriterden ödün vermeden başka bir kriterde iyileşme sağlanamayan çözümlerdir. Kriter uzayında bu etkin çözümlere karşılık gelen noktalar ise baskın noktalar olarak tanımlanır. Ehrgott ve Gandibleux [1], çizelgeleme problemlerini de içeren çok amaçlı bileşi problemleri üzerine inceleme yapmış, problem büyüklüğü arttıkça baskın noktaların sayısının üssel bir büyüme gösterdiğini belirtmiştir.

Çok amaçlı çizelgeleme problemleri için önerilen yaklaşımlar son dönemde giderek artmaktadır. Bu alanda ilk derleme çalışmaları içinde yer alan Dileepan ve Sen [2], tek makinalı çizelgeleme problemleri için önerilen yaklaşımları kriter sayısına göre ayırarak incelerken, Fry vd. [3] seçilen kriterlere göre bir sınıflandırma yapmıştır. Daha sonra, Nagar vd. [4] daha kapsamlı bir inceleme çalışması sunmaktadır. Bu çalışmada, yaklaşımlar problemin yapısı, kriter sayısı ve çözüm yaklaşımına göre sınıflandırılmaktadır. Benzer bir sınıflandırma yapan Eren ve Güner [5], çok amaçlı tek makinalı ve paralel makinalı çizelgeleme problemleri üzerine bir literatür taraması yapmıştır. Yen ve Wan [6] tarafından hazırlanan derleme

çalışmasında ise, iki amaçlı tek makinalı çizelgeleme problemleri için geliştirilen yöntemler çözüm yaklaşımına göre sınıflandırılmaktadır. Çok kriterli çizelgeleme problemleri üzerine daha detaylı incelemeler yapan T'kindt ve Billaut J-C [7] ve Hoogeveen [8], bu alandaki önemli bulguları sunmuştur.

Çok amaçlı problemlerin çözüm zorluğuna bağlı olarak daha çok iki kriterli çizelgeleme problemi üzerinde çalışılmıştır. Tablo 1, bu yaklaşımları seçilen kriterlerine göre sınıflandırmaktadır. İki amaçlı tek makinalı çizelgeleme problemleri için ilk yaklaşımlardan biri Van Wassenhove ve Gelders [9] tarafından önerilmiştir. Bu çalışmada, maksimum geç bitirme ve toplam akış zamanının en iyileştiği iki kriterli tek makinalı çizelgeleme problemi için tüm etkin çözümleri bulan bir algoritma geliştirilmiştir. Bu algoritma, temel olarak Smith [10] tarafından geliştirilen yöntemi kullanarak her adımda maksimum geç bitirme için verilen bir üst sınır kısıtı altında toplam akış zamanını en iyileyen çözümü bulmaktadır. Adım adım ilerleyen bu yaklaşım, tüm baskın noktaların bulunmasını garantiler. Benzer şekilde; Sen ve Gupta [11], Kondakçı vd. [12] ve Köksalan [13] maksimum geç bitirme ve toplam akış zamanını kriterlerinin en küçükleştiği çizelgeleme problemleri üzerine çalışmışlardır. Sen ve Gupta [11] ve Kondakçı vd. [12], iki kriterin herhangi bir doğrusal fonksiyonu olarak tanımlanan amaç fonksiyonu için en iyi çözümü üretebilen algoritma geliştirmişlerdir. Köksalan [13]

Tablo 1. İki kriterli tek makinalı çizelgeleme problemi için geliştirilen yaklaşımların kriterlerine göre sınıflandırılması
(Classification of single-machine bicriteria scheduling problems based on the criteria)

Kriterler	Çalışmalar
Toplam akış zamanı	Smith [10], Van Wassenhove ve Gelders [9], Sen ve Gupta
Maksimum geç bitirme	[11], Nelson vd. [15], Kondakçı vd. [12], Köksalan [13]
Toplam akış zamanı	Lin [14]
Toplam gecikme	
Geciken iş sayısı	
Maksimum geç bitirme	Nelson vd. [15], Chen ve Bulfin [16]
Toplam akış zamanı	Nelson vd. [15], Karasakal ve Köksalan [19]
Geciken iş sayısı	Köksalan ve Keha [20], Erenay vd. [17]
Toplam akış zamanı	Köksalan vd. [21], Köksalan [13],
Maksimum erken bitirme	Karasakal ve Köksalan [19], Köksalan ve Keha [20]
Geciken iş sayısı	
Maksimum erken bitirme	Güner vd. [18], Azizoğlu vd. [22]
Geciken iş sayısı	
Maksimum ağırlıklı geç bitirme	Huo vd. [24]
Geciken iş sayısı	
Ağırlıklı toplam erken bitirme	Wan ve Yen [25]
Ağırlıklı toplam akış zamanı	
Maksimum geç bitirme	Steiner ve Stephenson [26], He vd. [25]

ise iki kriterli tek makinalı çizelgeleme problemleri için baskın nokta kümesini yaklaşık olarak tanımlayan bir yüzey tanımlamıştır. Daha sonra tanımlanan bu yüzey üzerinde karar vericinin tercih ettiği bölgeleri belirleyerek, iyi etkin çözümler üretebilen sezgisel bir yöntem tasarlamıştır. Geliştirilen bu yöntem, toplam akış zamanı ve maksimum erken bitirme kriterlerinin en iyilendiği çizelgeleme problemi üzerinde de uygulanmıştır. Farklı olarak Lin [14], toplam akış zamanı ve toplam gecikme kriterlerinin en iyilendiği tek makinalı çizelgeleme problemleri için etkin çözümleri bulmayı hedefleyen dinamik programlamaya dayalı bir algoritma tasarlamıştır.

Tablo 1'de de gösterildiği gibi, iki kriterli çizelgeleme problemlerinde dikkate alınan bir diğer kriter geciken iş sayısıdır. İlk olarak, Nelson vd. [15], toplam akış zamanı, maksimum geç bitirme ve geciken iş sayısı kriterlerinin yer aldığı farklı iki kriterli problemler için dal ve sınır yöntemine dayalı algoritmalar geliştirmiştir. Daha sonra bu kriterlerin hepsinin en iyilendiği üç kriterli problem üzerine çalışmışlardır. Ancak, küçük problem üzerinde çalıştığı gösterilen yaklaşımlar, tüm etkin çizelgeleri bulmayı garantilememektedir. Benzer şekilde, Chen ve Bulfin [16] toplam geciken iş sayısı ve maksimum geç bitirme kriterlerini en iyilemeyi hedefleyen, dal ve sınır yöntemine dayalı sezgisel bir yöntem sunmaktadır. Erenay vd. [17] ise, tek makinalı çizelgeleme problemleri için akış zamanı ve geciken iş sayısını en küçükleyen dört farklı metasezgisel yöntem tasarlamıştır. Bu yaklaşımların ilk ikisinde dal ve sınır yönteminin bir uyarlaması olan ışın araması yaklaşımı ve Nelson vd. [15] tarafından önerilen dal ve sınır yöntemi ile birlikte kullanılırken, diğer yaklaşımlarda genetik algoritma ve tabu arama yöntemi kullanılmıştır. Farklı olarak, tek makinalı çizelgeleme problemleri için maksimum erken bitirmeyi en küçükleyen yaklaşımlar da tasarlanmıştır. Güner vd. [18], zamanında tamamlanması gereken işlerin kümesi verildiğinde maksimum erken bitirme kriterini en küçükleyen dal ve sınır yöntemine dayalı bir yaklaşım sunmaktadır. Karasakal ve Köksalan [19] ve Köksalan ve Keha [20] ise toplam akış zamanı ve geciken iş sayısını en iyilemeyi hedefleyen metasezgisel yöntemler geliştirmiş, bu yöntemi toplam akış zamanı ve maksimum erken bitirme kriterlerinin en küçüklendiği çizelgeleme problemleri için de uyarlamıştır. Karasakal ve Köksalan [19], benzetilmiş tavlama yöntemini kullanırken, Köksalan ve Keha [20] genetik algoritma tasarlamıştır. Köksalan vd. [21] tarafından önerilen sezgisel yöntemlerde ise toplam akış zamanı ve maksimum erken bitirme kriterlerinin en küçüklendiği tek makinalı çizelgeleme problemleri için küçük bir çözüm altkümesi bularak en iyi etkin çizelgeyi bulmayı hedefleyen bir algoritma geliştirilmiştir. Azizoğlu vd. [22] ise maksimum erken bitirme ve geciken iş sayısını en küçükleyen çizelgeleme problemleri için tüm etkin çözümleri bulan bir yöntem geliştirmiştir.

Shabtay ve Kaspi [23] tarafından da tartışıldığı gibi birçok çizelgeleme problemi sadece akış zamanına, gecikme veya erken bitirme sürelerine odaklanmıştır; işlerin önemi ve ağırlığı dikkate alınmamıştır. Bu nedenle, son dönemde tasarlanan yaklaşımlarda, işlerin ağırlığını gözeten kriterler

kullanılmaktadır. Örneğin, Huo vd. [24] toplam geciken iş sayısı ve maksimum ağırlıklı geç bitirme kriterlerinin en küçüklendiği tek makinalı çizelgeleme problemi üzerine çalışmış ve problemin çözüm zorluğunu üzerine incelemeler yapmışlardır. Bazı özel durumlar için kullanılabilir polinom zamanda çözülebilen algoritmaları tartışarak genel durum için hızlı çalışan bir sezgisel yöntem önermişlerdir. Wan ve Yen [25] ise, toplam geciken iş sayısının en küçük değerinde ağırlıklı toplam erken bitirme kriterini en iyileyen çözümü arayan sezgisel dal ve sınır yöntemi geliştirmişlerdir. He vd. [26], tek makinalı çizelgeleme problemleri için ağırlıklı toplam akış zamanını ve maksimum geç bitirme kriterlerini en iyileyen tüm etkin çözümleri bulan bir yaklaşım önermiş ancak bu çalışmada tüm işlerin işlem süresini aynı olarak varsaymışlardır. Benzer şekilde, Steiner ve Stephenson [27], ağırlıklı toplam akış zamanı ve maksimum geç bitirme kriterlerinin en iyilendiği iki kriterli tek makinalı çizelgeleme problemleri için bir yöntem geliştirmiştir. Küçük ölçekli problemlerde tüm baskın noktaları bulabilen bu yaklaşım, büyük ölçekli problemlerde baskın nokta kümesini temsil eden bir altküme bulacak şekilde tasarlanmıştır. Ancak, kriter uzayında düzgün dağılmış çözümler arayan bu yöntemde, temsilci kümesinin gerçek baskın nokta kümesine olan yakınlığını ölçülebilen bir kalite ölçüsü kullanılmamaktadır.

Baskın noktaların bulmanın çözüm zorluğuna bağlı olarak mevcut çalışmalarda, bu noktalara yakın noktalar bulabilen sezgisel/metasezgisel yöntemler ya da bu noktaların bir kısmını bulmayı hedefleyen algoritmalar kullanılmıştır. Ancak, bu çalışmalarda elde edilen çözümlerle gerçek baskın nokta kümesinin ne kadar iyi temsil edildiği değerlendirilmemiştir. Bu çalışmada, daha önceki çalışmalardan farklı olarak maksimum ağırlıklı geç bitirme ve ağırlıklı toplam akış zamanının en küçüklendiği iki kriterli çizelgeleme problemi için performans garantisi sunan bir algoritma geliştirilmiştir. Baskın noktaların hepsini bulan bu algoritma, mevcut çalışmalardan farklı olarak aynı zamanda istenilen kalite ölçüsünü sağlayan en küçük temsili bir altküme bulabilecek şekilde tasarlanmıştır. Bu sayede, tüm baskın nokta kümesini bulmanın zor olduğu büyük ölçekli problemlerde karar vericinin belirleyeceği kalite ölçüsüne göre çözümler üretmek mümkün olacaktır. Ayrıca, çizelgeleme yapılacak her işin ağırlığı karar verici tarafından seçilebilecek, ağırlığı yüksek işlerin önceliğine önem verilecektir. Bu çalışmada ilk olarak, ikinci kısımda iki kriterli problemimiz ve yaklaşımımız için genel bilgi ve tanımlar verilecektir. Daha sonra üçüncü kısımda geliştirilen yaklaşım sunulacak; dördüncü kısımda örnek bir problem üzerinde uygulaması gösterilecektir. Yapılan deneyler ve bulgular beşinci kısımda sunulacak, altıncı kısımda sonuçlar tartışılacaktır.

2. TANIMLAR VE GENEL BİLGİLER (DEFINITIONS AND BACKGROUND INFORMATION)

Maksimum ağırlıklı geç bitirme ve ağırlıklı toplam akış zamanının en küçüklendiği iki kriterli tek makinalı çizelgeleme problemi aşağıdaki şekilde tanımlanabilir:

(P)

$$\text{"Min"} z(\pi) = (z_1(\pi), z_2(\pi)) \quad (1)$$

Kısıtlar

$$\pi \in \Pi$$

Bu modelde, $\pi \in \mathbb{Z}^n$ vektörü n tane iş için sıralama vektörünü, $\Pi \in \mathbb{Z}^n$ olurlu uzay kümesini temsil etmektedir. $z_k(\pi)$, π sıralaması için k . kriterin aldığı değeri göstermektedir. Her zaman tüm kriterleri aynı anda en küçükleyen tek bir çözüm olmadığı için vektör minimizasyonu işlemi Eş. 1'de tırnak içinde gösterilmektedir. Karar vericinin kriterleri olarak maksimum ağırlıklı geç bitirme (WT_{max}) ve ağırlıklı toplam akış zamanı (WF) seçilmiş ve alttaki şekilde tanımlanmıştır:

$$z_1(\pi) = WF(\pi) = \sum_{i=1}^n w_i C_i(\pi) \quad (2)$$

$$z_2(\pi) = WT_{max}(\pi) = \max_{i=1, \dots, n} \max(0, w_i(C_i(\pi) - d_i)) \quad (3)$$

Burada, Eş. 2 ve 3'te numaralar verilen π sıralaması için, $C_i(\pi)$, i işinin tamamlanma zamanını ve w_i işin ağırlığını göstermektedir. Problemimizde, i işinin işlem zamanı ve teslim tarihi sırasıyla p_i ve d_i olarak tanımlanmıştır. Bu çalışmada, tüm bu parametrelerin tamsayı olduğu varsayılmıştır. Tamsayı olmaması durumunda tüm bu parametreler belirlenen bir katsayı ile çarpılarak tamsayıya dönüştürülebilir ve denk bu yeni problem çözülebilir.

Birinci kriteri eniyileyen çözüm işleri ağırlıklandırılmış en kısa çözüm süresi (WSPT) kuralı kullanılarak bulunur. Bu çözümde, işler p_i/w_i oranına göre küçükten büyüğe sıralanarak çizelgeleme elde edilir. İkinci kriteri eniyilemek için ise, ağırlıklandırılmış en erken teslim tarihi (WEDD) kuralı ile işler d_i/w_i oranına göre en küçükten büyüğe doğru sıralanarak çizelge elde edilir. Bu iki kriter aynı anda eniyilendiğinde ise bu kuralları aynı anda sağlayan bir çözüm yok ise etkin olan çizelgeleri bulmak hedeflenir. Baskın nokta ve etkin çözüm tanımları alta verilmiştir.

Tanım 1. Herhangi $\pi_1, \pi_2 \in \Pi$ çözümü için, $z_k(\pi_2) \leq z_k(\pi_1)$ $k = 1, 2$ ve en az bir amaç fonksiyonunda $z_k(\pi_2) < z_k(\pi_1)$ koşulları sağlanıyorsa, π_2 çözümünün π_1 çözümünü *baskıladığı* söylenir. Böyle bir π_2 çözümü yoksa, π_1 çözümüne *etkin çözüm* ve $z(\pi_1)$ noktasına *baskın nokta* denir.

Tanım 2. Herhangi $\pi_1, \pi_2 \in \Pi$ çözümü için, $z_k(\pi_2) < z_k(\pi_1)$ $k = 1, 2$ koşulu sağlanıyorsa, π_2 çözümünün π_1 çözümünü *tam baskıladığı* söylenir. Eğer böyle bir π_2 çözümü yoksa, π_1 çözümüne *zayıf etkin çözüm* ve $z(\pi_1)$ noktasına *zayıf baskın nokta* denir.

Tüm baskın noktaların kümesi Z_B olarak tanımlanmıştır. Toplam iş sayısının artmasıyla birlikte baskın noktaların sayısı hızla artabilir (bakınız Ehrgott ve Gandibleux [1]) ve tüm baskın noktaları bulmak büyük problemlerde pratik bir yaklaşım olmayabilir. Bu nedenle, baskın nokta kümesini iyi

temsil eden bir altküme, $Z_T \subseteq Z_B$, bulmak gerçek hayat büyüklüğündeki problemler için önemlidir. Üretilen bu altkümenin kalitesinin değerlendirilmesinde farklı performans ölçüleri tanımlanmıştır (bakınız, Sayın [28]; Wu ve Azarm [29]; Zitzler vd. [30], Faulkenberg ve Wiecek [31]). Bu performans ölçülerini kullanmada amaç, az sayıda çözüm ile tüm baskın noktaları iyi temsil eden, iyi kapsama özelliğine sahip bir altküme bulmaktır. Sayın [28] tarafından önerilen kapsama ölçüsü Tanım 3'te verilmiştir.

Tanım 3. Verilen temsili alt küme Z_T içinde, tüm $z \in Z_B$ baskın noktalar için en fazla $\alpha \geq 0$ uzaklığında en az bir temsilci $y \in Z_T$ varsa, $u(z, y) \leq \alpha$, Z_T alt kümesine Z_B 'nin α -temsili denir. Z_T alt kümesinin Z_B kümesini *kapsama hatası* $\alpha(Z_T) = \max_{z \in Z_B} \min_{y \in Z_T} u(z, y)$ şeklinde tanımlanır.

Tanım 3'te tanımlanan kapsama hatası, her bir baskın noktanın temsili alt küme tarafından ne kadar iyi temsil edildiğini ölçmek için kullanılmaktadır. Bu nedenle, verilen bir altkümenin kapsama hatasını hesaplamak için tüm baskın nokta kümesinin bilinmesi gerekmektedir. Kapsama hatası hesaplanırken, ilk olarak her bir baskın nokta, $z \in Z_B$, için temsili kümede yer alan en yakın nokta, $y \in Z_T$, temsilci olarak seçilmekte ve aralarındaki uzaklık, $u(z, y)$, ölçülerek z noktasının kapsama hatası bulunur. En kötü temsil edilen noktanın kapsama hatası tüm altkümenin kapsama hatası olarak değerlendirilir. Bu çalışmada, Sylva ve Crema [32], Masin ve Bukchin [33], ve Ceyhan vd. [34] çalışmalarında da olduğu gibi hesaplamalarda, $u(z, y) = \max_{k=1, 2} (y_k - z_k)$ uzaklık ölçüsü kullanılmaktadır.

Çok kriterli optimizasyon (ÇOP) problemlerinde baskın nokta üretmek kullanılan bir çok yöntem vardır. Bu yöntemlerden *Ağırlıklı Toplam* metodu, kriterleri pozitif ağırlıklı toplamını amaç fonksiyonu olarak seçmekte, ÇOP problemini tek kriterli optimizasyon problemine dönüştürmektedir. Ancak, bu yöntemle sadece *destekli baskın nokta* olarak tanımlanabilen çözümler üretilmekte, bu yöntemle üretilmesi mümkün olmayan *desteksiz baskın noktalar* bulunamamaktadır.

Destekli veya desteksiz tüm baskın noktaların bulunması için önerilen skalarlaştırma yöntemlerinden en önemlisi *epsilon-kısıtı yöntemi*dir. Bu yöntemde, bir kriter amaç fonksiyonu olarak seçilmekte ve diğer kriterler için üst sınır koyan kısıtlar eklenmektedir (Eş. 4 ve Eş. 5). İki kriterli (P) problemi için bu yöntemle baskın noktalar üretmek için (E^ϵ) problemi çözülmektedir.

$$(E^\epsilon) \quad \text{Min } z_1(\pi) \quad (4)$$

Kısıtlar

$$z_2(\pi) \leq \epsilon \quad (5)$$

$$x \in X$$

Bu modelde, ikinci kriterin amaç fonksiyonunda yer almaması nedeni ile üretilen baskın nokta ile birinci kriterde

aynı değere sahip ancak ikinci kriterde daha iyi olan başka alternatif çözümler bulunabilir. Bu nedenle, epsilon kısıtı yöntemi zayıf baskın olan ama baskın olmayan noktalar bulabilir. Baskın nokta bulmayı garantilemek amacı ile *modifiye edilmiş epsilon-kısıtı* yöntemi kullanılır. Bu yöntemde, Eş. 6'da yeterince küçük seçilen $\rho > 0$ değeri ile her adımda (M^ϵ) modeli çözülerek ve Eş. 7'de ikinci kriter için tanımlanan üst sınır sistematik şekilde değiştirilerek tüm baskın noktaları bulmak mümkündür.

$$(M^\epsilon) \\ \text{Min } z_1(\pi) + \rho z_2(\pi) \quad (6)$$

Kısıtlar

$$z_2(\pi) \leq \epsilon \\ x \in X \quad (7)$$

Van Wassenhove ve Gelders [9], toplam akış zamanı (F) ve maksimum geç bitirme (T_{max}) kriterlerinin enküçüklendiği tek makinalı çizelgeleme problemi için tüm baskın noktaları bulan bir yaklaşım önermiştir. $z_1(\pi) = F(\pi) = \sum_{i=1}^n C_i(\pi)$ ve $z_2(\pi) = T_{max}(\pi) = \max_{i=1, \dots, n} (0, (C_i(\pi) - d_i))$ olarak tanımlandığı bu problemde her adımda ikinci kriter için koyulan üst sınır ϵ olduğunda çözülen (P^ϵ) modeli alta tanımlanmıştır:

$$(P^\epsilon) \\ \text{Min } \sum_{i=1}^n C_i(\pi) + \rho T_{max}(\pi) \quad (8)$$

Kısıtlar

$$T_{max}(\pi) \leq \epsilon \\ \pi \in \Pi \quad (9)$$

Modifiye edilmiş epsilon-kısıtı yöntemini kullanan bu yaklaşımda ilk olarak ikinci kriter için bir üst sınır konmamakta ve toplam akış zamanını eniyileyen çözüm bulunmaktadır. SPT (en küçük işlem sırası) kuralına göre belirlenen bu çizelge için alternatifler olması durumunda teslim tarihleri dikkate alınarak sıralama yapılmaktadır. Elde edilen bu baskın noktaya karşılık gelen çizelgenin maksimum geç bitirme, $T_{max}(\pi_{SPT})$, değerine göre bir sonraki iterasyon için $\epsilon < T_{max}(\pi_{SPT})$ olacak şekilde yeni bir üst sınır, ϵ , belirlenmektedir: Burada bir sonraki çözümde T_{max} değerinin iyileştirmek için F değerinden ödün vererek yeni bir baskın nokta bulunması garantilenmektedir. Parametrelerin tamsayı olarak seçildiği Van Wassenhove ve Gelders [9] algoritmasında $\epsilon = T_{max}(\pi_{SPT}) - 1$ olarak seçilmektedir. Bir sonraki iterasyonda, yeni belirlenen üst sınırın tanımlanmasıyla kurulan (P^ϵ) modeli çözülür. Aynı şekilde elde edilen çizelgenin, π_ϵ , geç bitirme değeri, $T_{max}(\pi_\epsilon)$ hesaplanır ve bir sonraki iterasyon için üst sınır değeri güncellenir: $\epsilon = T_{max}(\pi_\epsilon) - 1$ ve (P^{ϵ}) modeli çözülür. Bu işlemler (P^ϵ) modeli olursuz olana kadar devam eder. Her iterasyonda güncellenen (P^ϵ) modelini çözümünde Van Wassenhove ve Gelders [9], Smith'in geri yönlü çizelgeleme kuralını modifiye ederek kullanmaktadır. Öyle ki, maksimum geç bitirme (T_{max}) kriteri için üst sınır ϵ

olarak tanımlandığından tüm teslim tarihleri ϵ kadar ötelenerek revize edilir: $d'_i = d_i + \epsilon$. Smith'in geri yönlü çizelgeleme kuralı revize edilerek çözülür:

- i. $R = \sum_{i=1}^n p_i$, $N = \{1, \dots, n\}$, $s = n$.
- ii. Henüz çizelgelenmemiş, $i \in N$, ve $d'_i \geq R$ koşulunu sağlayan işler içerisinde en büyük işlem süresine sahip i^* işi, $p_{i^*} \geq p_i$, s pozisyonuna atanır. Belirtilen koşulları sağlayan birden fazla seçenek olması durumunda teslim tarihi büyük olan iş seçilir.
- iii. $R = R - p_{i^*}$ ve $N = N - \{i^*\}$, $s = s - 1$ şekilde güncellenir. $s > 0$ için (ii) adımı tekrarlanır.

3. GELİŞTİRİLEN ALGORİTMA (THE DEVELOPED ALGORITHM)

Bu çalışmada, ağırlıklı toplam akış zamanının ve maksimum ağırlıklı geç bitirmenin eniyilendiği iki kriterli tek makinalı çizelgeleme problemi (P) için bir algoritma geliştirilmiştir. Tüm baskın noktaları bulabilen bu algoritma, aynı zamanda tüm baskın nokta kümesini istenilen kalite seviyesinde temsil eden altküme de bulabilecek şekilde tasarlanmıştır. Öyle ki, karar verici tarafından tolere edilebilecek kapsama hatası α olarak verilmişse, $\alpha(Z_T) \leq \alpha$ koşulunu sağlayan bir altküme $Z_T \subseteq Z_B$ bulmak garantilenmektedir.

İlk kriteri, ağırlıklı toplam akış zamanını, en iyileyen etkin çözüm ile başlayan bu algoritma daha sonra her adımda bir önceki çözüme göre ikinci kriteri, maksimum ağırlıklı geç bitirme değeri, daha iyi olan yeni bir etkin çözüm bulmaktadır. İkinci kriter değeri daha da azaltmanın mümkün olmadığı zaman da algoritma durur. Önerme 1'de verildiği üzere ilk çözüm WSPT kuralı uygulanılarak elde edilir.

Önerme 1. WSPT kuralını sağlayan etkin bir çözüm vardır.

İspat. Her $i \in N$ için p_i/w_i değerleri birbirinden farklı ise WSPT kuralını sağlayan tek bir çözüm vardır ve bu çözüm ağırlıklı toplam akış zamanının en iyileyen çözümdür: $z_1(\pi_{WSPT}) = \min_{\pi \in \Pi} \sum_{i=1}^n w_i C_i(\pi)$. Bu nedenle, $z_1(\pi) < z_1(\pi_{WSPT})$ kısıtını sağlayan başka bir $\pi \in \Pi$ çözüm olmadığı için, $z(\pi_{WSPT})$ noktasını baskılayan da başka çözüm yoktur ve π_{WSPT} etkin bir çözümdür (bakınız Tanım 1 ve 2). Bazı i ve j işleri için $p_i/w_i = p_j/w_j$ oranı birbirine eşit ise, WSPT kuralını sağlayan birden fazla çözüm vardır. Bu çözümler içerisinde yer alan ve eşit orana sahip olan işlerin kendi aralarında sıralanmasında WEDD kuralı kullanıldığında elde edilen çözüm, π_{WSPT} , etkindir. Çünkü diğer alternatif çözümler için, $z_1(\pi_{WSPT}) = z_1(\pi)$ iken WEDD kuralı uygulandığından $z_2(\pi_{WSPT}) \leq z_2(\pi)$ kısıtı sağlanmaktadır ve $z(\pi_{WSPT})$ noktasını baskılayan da başka çözüm yoktur.

WSPT kuralı uygulayarak ilk etkin çözümünü bulan bu algoritma daha sonra modifiye edilmiş epsilon kısıtı yöntemini kullanmakta ve her adımda bir öncekine göre ilk kriterden ödün vererek ikinci kriteri iyileştiren yeni çözümler üretmektedir. İkinci kriter değeri için istenilen üst sınır ϵ

olarak verildiğinde yeni bir etkin çözüm üretmek için çözülecek model Önerme 2’de sunulmuştur.

Önerme 2. $\varepsilon \geq WT_{max}(\pi_{WEDD})$ ve $0 < \rho < \frac{1}{\varepsilon}$ yeterince küçük pozitif bir sayı ise (WP^ε) modeli $WT_{max}(\pi) \leq \varepsilon$ koşulunu sağlayan çözümler içerisinde en küçük $\sum_{i=1}^n w_i C_i(\pi)$ değerine sahip etkin çözümü üretir.

$$(WP^\varepsilon) \quad \text{Min } \sum_{i=1}^n w_i C_i(\pi) + \rho WT_{max}(\pi) \quad (10)$$

Kısıtlar

$$WT_{max}(\pi) \leq \varepsilon \quad (11)$$

$$\pi \in \Pi$$

İspat. (WP^ε) modelinde Eş. 11’de $\varepsilon \geq WT_{max}(\pi_{WEDD})$ olarak seçildiğinden (WP^ε) problemi için en az bir olurlu çözüm vardır. (WP^ε) modelinin çözümüne karşılık gelen optimal çözümü $\pi_{(WP^\varepsilon)}$ olarak tanımlarsak, $\pi_{(WP^\varepsilon)}$ çözümünün etkin bir çözüm olduğunu Olmayana Ergi yöntemi ile gösterebiliriz. Eğer $\pi_{(WP^\varepsilon)}$ etkin bir çözüm değilse, onu baskılayan (Eş. 12 ve 13’te yer alan eşitsizliği ve en az bir tanesinde kesin eşitsizliği sağlayan) başka bir $\pi \neq \pi_{(WP^\varepsilon)}$ çözümü vardır.

$$\sum_{i=1}^n w_i C_i(\pi) \leq \sum_{i=1}^n w_i C_i(\pi_{(WP^\varepsilon)}) \quad (12)$$

$$WT_{max}(\pi) \leq WT_{max}(\pi_{(WP^\varepsilon)}) \quad (13)$$

Bu durumda $\sum_{i=1}^n w_i C_i(\pi) + \rho WT_{max}(\pi) \leq \sum_{i=1}^n w_i C_i(\pi_{(WP^\varepsilon)}) + \rho WT_{max}(\pi_{(WP^\varepsilon)})$ da sağlanmalı bu da $\pi_{(WP^\varepsilon)}$ optimal çözüm olduğu bilgisi ile çelişmektedir. Bu çelişki $\pi_{(WP^\varepsilon)}$ çözümünün etkin olduğunu gösterir. $\pi_{(WP^\varepsilon)}$ çözümünün $WT_{max}(\pi) \leq \varepsilon$ koşulunu sağlayan diğer etkin çözümler içerisinde en küçük ağırlıklı toplam akış zamanına sahip çözüm olduğunu göstermek amacı ile olmayana ergi yöntemini kullanırsak $\sum_{i=1}^n w_i C_i(\pi) < \sum_{i=1}^n w_i C_i(\pi_{(WP^\varepsilon)})$ ve $WT_{max}(\pi) \leq \varepsilon$ koşullarını sağlayan başka bir etkin çözüm olmalıdır. Bu durumda,

(i) π ve $\pi_{(WP^\varepsilon)}$ ikisi de etkin çözüm olduğu için

$$WT_{max}(\pi) > WT_{max}(\pi_{(WP^\varepsilon)}) \quad (14)$$

(ii) $\pi_{(WP^\varepsilon)}$ optimal çözüm olduğu için

$$\sum_{i=1}^n w_i C_i(\pi_{(WP^\varepsilon)}) + \rho WT_{max}(\pi_{(WP^\varepsilon)}) \leq \sum_{i=1}^n w_i C_i(\pi) + \rho WT_{max}(\pi) \quad (15)$$

kısıtları sağlanmalıdır. Eş. 14 ve 15’te yer alan bu eşitsizlikleri düzenlersek, tüm parametreler tamsayı olarak tanımlandığından $\rho \geq \frac{\sum_{i=1}^n w_i C_i(\pi_{(WP^\varepsilon)}) - \sum_{i=1}^n w_i C_i(\pi)}{[WT_{max}(\pi) - WT_{max}(\pi_{(WP^\varepsilon)})]} \geq \frac{1}{\varepsilon}$ koşulu sağlanmalıdır ve bu önermede yer alan $0 < \rho < \frac{1}{\varepsilon}$ ile

çelişmektedir. Bu çelişki, ρ yeterince küçük seçildiği durumda, (WP^ε) modelinin $WT_{max}(\pi) \leq \varepsilon$ koşulunu sağlayan çözümler içerisinde en küçük $\sum_{i=1}^n w_i C_i(\pi)$ değerine sahip etkin çözümü ürettiğini gösterir.

Verilen ε değeri için tanımlanan (WP^ε) modelin çözümünde de Smith’in geri yönlü çizelgeleme kuralı [9] revize edilecektir. Öyle ki,

(1) Maksimum ağırlıklı geç bitirme değeri için üst sınır ε olarak verildiğinde her $i \in N$ işi için $C_i(\pi) \leq d_i + (\varepsilon/w_i)$ kısıtı sağlanmalıdır. Bu durumda, her $i \in N$ işi için maksimum gecikme miktarı ε/w_i kadar olup tüm teslim tarihleri ε/w_i kadar ötelenerek revize edilir: $d'_i = d_i + (\varepsilon/w_i)$.

(2) ağırlıklı toplam akış zamanı en iyileştiğinden belirtilen kısıtları sağlayan işler içerisinde işlem süresi / ağırlık oranı (p_i/w_i) dikkate alınarak seçim yapılır.

Mevcut algoritmalarından farklı olarak istenilen kalite seviyesine sahip bir altküme üretebilmek amacı ile her adımda ikinci kriter için üst sınır, ε , tolere edilebilecek kapsama hatası değerine, α , göre belirlenmektedir. Önerme 3, Temsili Çizelgeleme kümesi Bulma Algoritması (TÇBA) olarak tanımlanan bu algoritmanın $\alpha(Z_T) \leq \alpha$ koşulunu sağlayan bir altküme $Z_T \subseteq Z_B$ bulmayı garantilediğini göstermektedir.

3.1. Temsili çizelgeleme kümesi bulma algoritması (An Algorithm to find a set of representative schedules)

Adım 1. Karar verici tarafından işlerin ağırlığı, w_i $i \in N$, ve üretilecek temsili kümenin kapsama hatası, α , belirlenir. $\varepsilon = (\sum_{i=1}^n p_i) \left(\max_{i \in N} w_i \right)$, ve $d'_i = d_i + (\varepsilon/w_i)$, $N' = N$ olarak ilk değerleri atanır. Temsili baskın nokta kümesi $Z_T = \emptyset$, temsilci sayısı $r = 0$ olarak tanımlanır.

Adım 2. (WP^ε) modeli Smith’in geri yönlü çizelgeleme kuralı ile çözülür:

Adım 2.1. $R = \sum_{i=1}^n p_i$, $N = \{1, \dots, n\}$, ve $s = n$ olarak ilk değerleri belirlenir.

Adım 2.2. Henüz çizelgelenmemiş, $d'_i \geq R$ koşulunu sağlayan N' kümesindeki işler içerisinde en büyük işlem süresi / ağırlık oranına sahip i^* işi, $i^* = \arg \max_{i \in N'} \frac{p_i}{w_i}$, s pozisyonuna atanır, $\pi_s = i^*$. Belirtilen koşulları sağlayan birden fazla seçenek olması durumunda d_i/w_i oranı daha büyük olan iş seçilir. Adım 2.3 ile devam edilir.

İlgili pozisyona atanacak iş olmaması durumunda, $N' = \emptyset$, ise 3. adıma geçilir.

Adım 2.3. $s = s - 1$, $R = R - p_{i^*}$ ve $N = N - \{i^*\}$ şeklinde güncellenir. Daha sonra $N' = \{i \in N : d'_i \geq R\}$ olarak belirlenir.

Güncellemelerden sonra $s > 0$ ise 2.2 adımına gidilir. $s = 0$ ise $r = r + 1$ olarak değiştirilir. Bulunan sıralama vektörüne karşılık gelen çözüm, $y^r = (\sum_{i=1}^n w_i C_i(\pi), WT_{max}(\pi))$, temsili kümeye eklenir: $Z_T = Z_T \cup \{y^r\}$. Bulunan sıralamaya bağlı olarak bir sonraki iterasyon için $\varepsilon = WT_{max}(\pi) - \alpha$ değeri belirlenir ve ona bağlı olarak her $i \in N$ işi için $d'_i = d_i + (\varepsilon/w_i)$ değerleri güncellenir. Adım 2 tekrarlanır.

Adım 3. Algoritma durdurulur.

Önerme 3. $\alpha > 0$ için TÇBA tarafından üretilen $Z_T = \{y^1, \dots, y^r\}$ altkümesinde yer alan tüm ardışık çözümler için $y_1^m < y_1^{m+1}$ ve $y_2^m > y_2^{m+1}$ $m = 1, \dots, r - 1$ koşulları sağlanır.

İspat. TÇBA, herhangi bir $m \in \{1, \dots, r - 1\}$ iterasyonunda elde edilen y^m çözümünü kullanarak Adım 2.3'te ikinci kriter için üst sınır olarak tanımlanan $\varepsilon = y_2^m - \alpha$ değerini günceller. Bu nedenle, bu ε değeri ile çözülen (WP^ε) modelinin çözümü olan y^{m+1} için $y_2^{m+1} \leq \varepsilon$ kısıtı, dolayısıyla $y_2^{m+1} \leq y_2^m - \alpha$ eşitsizliği sağlanır. Bu nedenle, $\alpha > 0$ için $y_2^{m+1} < y_2^m$ sağlanmaktadır. Önerme 2'de gösterildiği gibi, (WP^ε) modeli ikinci kriter için ε koşulunu sağlayan çözümler içerisinde, birinci kriter değeri en küçük etkin çözümü üretir. $y_2^{m+1} < y_2^m$ koşulu her $m \in \{1, \dots, r - 1\}$ için sağlandığından ve ilgili modellerin üst sınırları, ε_{m+1} ve ε_m , arasında da $\varepsilon_{m+1} < \varepsilon_m$ ilişkisi vardır: $y_2^{m+1} \leq \varepsilon_{m+1} = y_2^m - \alpha < y_2^m \leq \varepsilon_m$. Üst sınırın azalmasıyla olurlu çözüm kümesi daralmakta bu nedenle birinci kriter değeri kötüleşerek ya da sabit kalarak ilerler. Önerme 2'de, (WP^ε) modeli ile üretilen çözümler olan y^m ve y^{m+1} çözümleri için $y^m, y^{m+1} \in Z_B$ gösterildiğinden bu iki çözüm birbirini baskılamaz ve bu nedenle $y_2^{m+1} < y_2^m$ koşulunun yanında $y_1^{m+1} > y_1^m$ koşulu da sağlanır.

Önerme 4. $u(z, y) = \max_{k=1, \dots, p} (z_k - y_k)$ uzaklık ölçüsü kullanılıyorsa, TÇBA tarafından üretilen $Z_T = \{y^1, \dots, y^r\}$ alt kümesinin Z_B kümesini kapsama hatası en fazla α 'dır: $\alpha(Z_T) \leq \alpha$.

İspat. Olmaya ergi yöntemi ile $\alpha(Z_T) > \alpha$ olduğunu varsayalım. Bu durumda, en az bir baskın çözüm için $z' \in Z_B$, en fazla α uzaklığında yer alan Z_T içinde bir temsilci bulunmamaktadır. Bu durumda, bu baskın çözüm için tüm temsilciler, $y \in Z_T$, $u(z', y) > \alpha$ uzaklığındadır. $u(z', y^m) = \max_{k=1, 2} (y_k^m - z'_k)$ olarak tanımlandığından her temsilci için en az bir kriterde Eş. 16 sağlanmalıdır.

$$y_k^m \geq z'_k + \alpha \text{ tüm } m = 1, \dots, r - 1 \quad (16)$$

TÇBA, çözümleri ikinci kriter göre azalan ilk kriter göre artan sırada ürettiğinden (bakınız Önerme 3) $y_1^m < y_1^{m+1}$ ve $y_2^m > y_2^{m+1}$ koşulları sağlanmaktadır. Bu nedenle, öyle bir m temsilcisi olmalıdır ki, bu temsilci için, Eş. 17 ve 18 sağlanmalıdır.

$$y_1^1 < y_1^m < z'_k + \alpha < y_1^{m+1} < y_1^r \quad (17)$$

$$y_2^1 > y_2^m > z'_k + \alpha > y_2^{m+1} > y_2^r \quad (18)$$

Bu koşullar altında, $z' \in Z_B$ çözümü $(m + 1)$. iterasyonda çözülen (WP^ε) modeli için $z'_k < \varepsilon = y_2^m - \alpha$ kısıtı sağladığından olurlu bir çözümdür ve de $z'_k < y_1^{m+1} - \alpha$ kısıtı sağladığından y^m çözümünden daha iyi ilk kriter değerine sahiptir. Önerme 2'de de gösterildiği gibi bu ε değeri ile çözülen (WP^ε) modelinin çözümü olan y^{m+1} olurlu çözümler içinde en küçük ilk kriter değerine sahip çözüm olarak tanımlanmıştır ve $z'_k < y_1^{m+1} - \alpha$ ile çelişmektedir. Bu çelişki, bulunamayan tüm baskın noktaların temsilcilere en fazla α uzaklığında yer aldığını ispatlamaktadır.

3.2. Geliştirilen algoritmanın örnek üzerinde uygulanması (Application of the algorithm on an example problem)

Bu bölümde, TÇBA'nın nasıl çalıştığı, bulunan çözümler ve özellikleri Tablo 2'de sunulan örnek üzerinde gösterilmiştir.

Iterasyon 1, Adım 1. $\alpha = 10$, $\sum_{i=1}^n p_i = 69$, $\varepsilon = (\sum_{i=1}^n p_i) \left(\max_{i \in N} w_i \right) = 690$, $d'_i = d_i + (\varepsilon/w_i)$, $Z_T = \emptyset$, $r = 0$

Iterasyon 1, Adım 2. (WP^ε) modeli Smith'in geri yönlü çizelgeleme kuralı ile çözümler işlerin sıralanması (π_1) bulunur (bakınız Tablo 3). Birinci kriteri eniyileyen bu çözüm WSPT kuralı ile işleri $\frac{p_i}{w_i}$ oranlarına göre sıralayarak da bulunabilir.

$$z_1(\pi_1) = \sum_{i=1}^{10} w_i C_i(\pi_1) = 1253 \quad (19)$$

$$z_2(\pi_1) = \max_{i=1, \dots, 10} \max(0, w_i(C_i(\pi_1) - d_i)) = 144 \quad (20)$$

$$y^2 = (1253, 144) \quad (21)$$

$$Z_T = \{y^1\} \quad (22)$$

Bir sonraki iterasyon için $\varepsilon = WT_{max}(\pi_1) - \alpha = z_2(\pi_1) - 10 = 134$ değeri belirlenir ve ona bağlı olarak her $i \in N$ işi için $d'_i = d_i + (\varepsilon/w_i)$ değerleri güncellenir. Adım 2 tekrarlanır.

Tablo 2. Örnek problem için parametre değerleri (The values of the parameters for the example problem)

İşler	1	2	3	4	5	6	7	8	9	10
İşlem süresi (p_i)	9	5	7	4	8	10	4	7	6	9
Teslim Tarihi (d_i)	9	38	43	27	17	10	38	32	17	46
İşin Ağırlığı (w_i)	4	7	9	9	3	1	7	9	1	5

İterasyon 2, Adım 2. (WP^ε) modeli ($\varepsilon = 134$) Smith'in geri yönlü çizelgeleme kuralı ile çözümlenerek işlerin sıralanması (π_2) bulunur (bakınız Tablo 4).

$$z_1(\pi_2) = \sum_{i=1}^{10} w_i C_i(\pi_2) = 1262 \quad (23)$$

$$z_2(\pi_2) =$$

$$\max_{i=1, \dots, 10} \max(0, w_i(C_i(\pi_2) - d_i)) = 108 \quad (24)$$

$$y^2 = (1262, 108) \quad (25)$$

$$Z_T = \{y^1, y^2\} \quad (26)$$

$$\varepsilon = WT_{max}(\pi_2) - \alpha = z_2(\pi_2) - 10 = 98 \quad (27)$$

İterasyon 3, Adım 2. (WP^ε) modeli ($\varepsilon = 98$) Smith'in geri yönlü çizelgeleme kuralı ile çözümlenerek işlerin sıralanması (π_3) bulunur (bakınız Tablo 5).

$$z_1(\pi_3) = \sum_{i=1}^{10} w_i C_i(\pi_3) = 1328 \quad (28)$$

$$z_2(\pi_3) = \max_{i=1, \dots, 10} \max(0, w_i(C_i(\pi_3) - d_i)) = 81 \quad (29)$$

$$y^3 = (1328, 81) \quad (30)$$

$$Z_T = \{y^1, y^2, y^3\} \quad (31)$$

$$\varepsilon = WT_{max}(\pi_3) - \alpha = z_2(\pi_3) - 10 = 71 \quad (32)$$

İterasyon 4, Adım 2. (WP^ε) modeli ($\varepsilon = 71$) Smith'in geri yönlü çizelgeleme kuralı ile çözümlenerek işlerin sıralanması (π_4) bulunur (bakınız Tablo 6).

$$z_1(\pi_4) = \sum_{i=1}^{10} w_i C_i(\pi_4) = 1432 \quad (33)$$

$$z_2(\pi_4) = \max_{i=1, \dots, 10} \max(0, w_i(C_i(\pi_4) - d_i)) = 60 \quad (34)$$

$$y^4 = (1432, 60) \quad (35)$$

$$Z_T = \{y^1, y^2, y^3, y^4\} \quad (36)$$

$$\varepsilon = WT_{max}(\pi_4) - \alpha = z_2(\pi_4) - 10 = 50 \quad (37)$$

Tablo 3. Örnek problemimiz için bulunan birinci çözüm (The first solution generated by the algorithm)

İşlerin sıralanması (π_1)	4	7	2	8	3	10	1	5	9	6
İşlem süresi (p_i)	4	4	5	7	7	9	9	8	6	10
İşin tamamlanma zamanı $C_i(\pi_1)$	4	8	13	20	27	36	45	53	59	69
İşin Ağırlığı (w_i)	9	7	7	9	9	5	4	3	1	1
Teslim Tarihi (d_i)	27	38	38	32	43	46	9	17	17	10
Ötelenmiş Teslim Tarihi $d'_i = d_i + (\varepsilon/w_i)$	104	137	137	109	120	184	182	247	707	700
$\frac{p_i}{w_i}$ oranı	0,44	0,57	0,71	0,78	0,78	1,80	2,25	2,67	6,00	10,00
$\frac{d_i}{w_i}$ oranı	3,00	5,43	5,43	3,56	4,78	9,20	2,25	5,67	17,00	10,00
İşin geç bitirme süresi $\max(0, (C_i(\pi_1) - d_i))$	0	0	0	0	0	0	36	36	42	59
İşin ağırlıklı geç bitirme süresi $\max(0, w_i(C_i(\pi_1) - d_i))$	0	0	0	0	0	0	144	108	42	59

Tablo 4. Örnek problemimiz için bulunan ikinci çözüm (The second solution generated by the algorithm)

İşlerin sıralanması (π_2)	4	7	2	8	3	1	10	5	9	6
İşlem süresi (p_i)	4	4	5	7	7	9	9	8	6	10
İşin tamamlanma zamanı $C_i(\pi_2)$	4	8	13	20	27	36	45	53	59	69
İşin Ağırlığı (w_i)	9	7	7	9	9	4	5	3	1	1
Teslim Tarihi (d_i)	27	38	38	32	43	9	46	17	17	10
Ötelenmiş Teslim Tarihi $d'_i = d_i + (\varepsilon/w_i)$	41	57	57	46	57	42	72	61	151	144
$\frac{p_i}{w_i}$ oranı	0,44	0,57	0,71	0,78	0,78	2,25	1,80	2,67	6,00	10,00
$\frac{d_i}{w_i}$ oranı	3,00	5,43	5,43	3,56	4,78	2,25	9,20	5,67	17,00	10,00
İşin geç bitirme süresi $\max(0, (C_i(\pi_2) - d_i))$	0	0	0	0	0	27	0	36	42	59
İşin ağırlıklı geç bitirme süresi $\max(0, w_i(C_i(\pi_2) - d_i))$	0	0	0	0	0	108	0	108	42	59

İterasyon 5, Adım 2. (WP^ε) modeli ($\varepsilon = 50$) Smith'in geri yönlü çizelgeleme kuralı ile olurlu bir çözüm bulunamaz. Adım 3 ile devam edilir.

İterasyon 5, Adım 3. Algoritma durdurulur. Bulunan temsili alt küme: $Z_T = \{y^1, y^2, y^3, y^4\}$.

Tablo 7'de de gösterildiği gibi 10-işli bu küçük örnekte toplam 7 etkin çizelgeleme çözümü var iken TÇBA $\alpha = 10$ değeri ile çözüldüğünde bu çizelgelerin dört tanesi $\{z^1, z^2, z^3, z^5\}$ üretilmektedir. Bu örnekte, bulunamayan baskın noktalardan $z^4 = (1379, 80)$ çözümü y^3 temsilcisi tarafından, $z^6 = (1483, 59)$ ve $z^7 = (1487, 53)$ ise y^4

temsilcisi tarafından temsil edilmektedir. En kötü temsil edilen baskın nokta z^7 için temsil hatası şöyle hesaplanabilir: $u(z^7, y^4) = \max_{k=1,2} (y_k^4 - z_k^7) = \max(1432 - 1487, 60 - 53) = 7$. Önerme 4'te de gösterildiği gibi $Z_T = \{y^1, y^2, y^3, y^4\}$ alt kümesinin Z_B kümesini kapsama hatası, en fazla $\alpha = 10$ kadardır.

4. DENEYLER (COMPUTATIONAL EXPERIMENTS)

Deneylerimizde kullanılan problemler temel olarak dört farklı parametre setine göre tanımlanmıştır: toplam iş sayısı (n), teslim tarihi aralık faktörü (ρ), gecikme faktörü (τ) ve kapsama hata yüzdesi (α).

Tablo 5. Örnek problemimiz için bulunan üçüncü çözüm (The third solution generated by the algorithm)

İşlerin sıralanması (π_3)	4	7	2	8	1	3	5	10	9	6
İşlem süresi (p_i)	4	4	5	7	9	7	8	9	6	10
İşin tamamlanma zamanı $C_i(\pi_3)$	4	8	13	20	29	36	44	53	59	69
İşin Ağırlığı (w_i)	9	7	7	9	4	9	3	5	1	1
Teslim Tarihi (d_i)	27	38	38	32	9	43	17	46	17	10
Ötelenmiş Teslim Tarihi $d'_i = d_i + (\varepsilon/w_i)$	37	52	52	42	33	53	49	65	115	108
$\frac{p_i}{w_i}$ oranı	0,44	0,57	0,71	0,78	2,25	0,78	2,67	1,80	6,00	10,00
$\frac{d_i}{w_i}$ oranı	3,00	5,43	5,43	3,56	2,25	4,78	5,67	9,20	17,00	10,00
İşin geç bitirme süresi $\max(0, (C_i(\pi_3) - d_i))$	0	0	0	0	20	0	27	7	42	59
İşin ağırlıklı geç bitirme süresi $\max(0, w_i(C_i(\pi_3) - d_i))$	0	0	0	0	80	0	81	35	42	59

Tablo 6. Örnek problemimiz için bulunan dördüncü çözüm (The fourth solution generated by the algorithm)

İşlerin sıralanması (π_4)	4	7	2	1	8	5	3	10	9	6
İşlem süresi (p_i)	4	4	5	9	7	8	7	9	6	10
İşin tamamlanma zamanı $C_i(\pi_4)$	4	8	13	22	29	37	44	53	59	69
İşin Ağırlığı (w_i)	9	7	7	4	9	3	9	5	1	1
Teslim Tarihi (d_i)	27	38	38	9	32	17	43	46	17	10
Ötelenmiş Teslim Tarihi $d'_i = d_i + (\varepsilon/w_i)$	34,9	48,1	48,1	26,8	39,9	40,7	50,9	60,2	88,0	81,0
$\frac{p_i}{w_i}$ oranı	0,44	0,57	0,71	2,25	0,78	2,67	0,78	1,80	6,00	10,00
$\frac{d_i}{w_i}$ oranı	3,00	5,43	5,43	2,25	3,56	5,67	4,78	9,20	17,00	10,00
İşin geç bitirme süresi $\max(0, (C_i(\pi_4) - d_i))$	0	0	0	13	0	20	1	7	42	59
İşin ağırlıklı geç bitirme süresi $\max(0, w_i(C_i(\pi_4) - d_i))$	0	0	0	52	0	60	9	35	42	59

Tablo 7. Örnek problemimiz için tüm etkin çizelgeleme çözümleri
(The set of all efficient schedules for the example problem)

Çözüm	İşlerin sıralanması (π)										$z_1(\pi)$	$z_2(\pi)$
z^1	4	7	2	8	3	10	1	5	9	6	1253	144
z^2	4	7	2	8	3	1	10	5	9	6	1262	108
z^3	4	7	2	8	1	3	5	10	9	6	1328	81
z^4	4	7	2	8	1	5	3	10	9	6	1379	80
z^5	4	7	2	1	8	5	3	10	9	6	1432	60
z^6	4	7	2	1	5	8	3	10	9	6	1483	59
z^7	4	7	2	1	5	8	3	10	6	9	1487	53

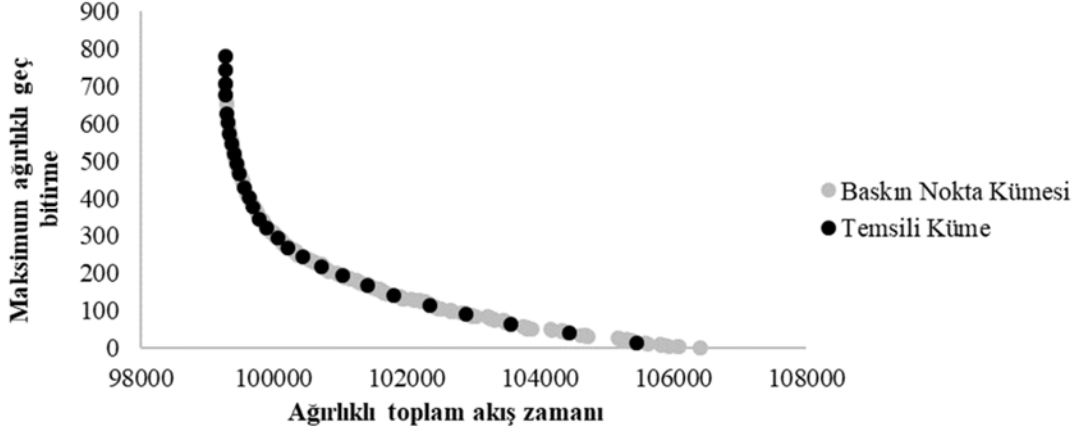
Öyle ki $\rho, \tau \in \{0,2; 0,4; 0,6; 0,8; 1,0\}$ ve $n \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$ ve $\alpha = \{1, 25, 50, 75, 100\}$ değerleri kullanılmıştır. Bu problemlerde, işlem süreleri $p_i \in [1,10]$ veya $p_i \in [1,50]$ aralığından, ağırlıklar $w_i \in [1,10]$ aralığından, teslim süreleri ise $[\sum_{i=1}^n p_i(1 - \tau - \rho/2), \sum_{i=1}^n p_i(1 - \tau + \rho/2)]$ aralığından ayrık tekdüze dağılıma göre üretilmiştir. (bakınız Huo vd. [24], Baker ve Martin [35], Rinnooy vd. [36]; Ow ve Morton [37]). Bu şekilde toplam 500 farklı problem 5

farklı kapsama hata yüzdesi kullanılarak 2500 farklı örnek çözülmüştür. Algoritma, Microsoft Visual Studio 2010 C++ programla dilini ile kullanılarak kodlanmış, 64-bit Microsoft Windows 7 Professional yazılımı yüklü Intel (R) Core (TM) i7-4790 CPU @ 3.60GHz 8.00 GB RAM özelliklerine sahip bilgisayarda deneyler yapılmıştır. Sonuçlar Tablo 8’de sunulmuştur. Şekil 1 -Şekil 3’te de gösterildiği gibi, istenilen kalite seviyesi arttıkça (istenilen kapsama değeri, α , azaldıkça) bulunan temsili kümenin büyüklüğü ve

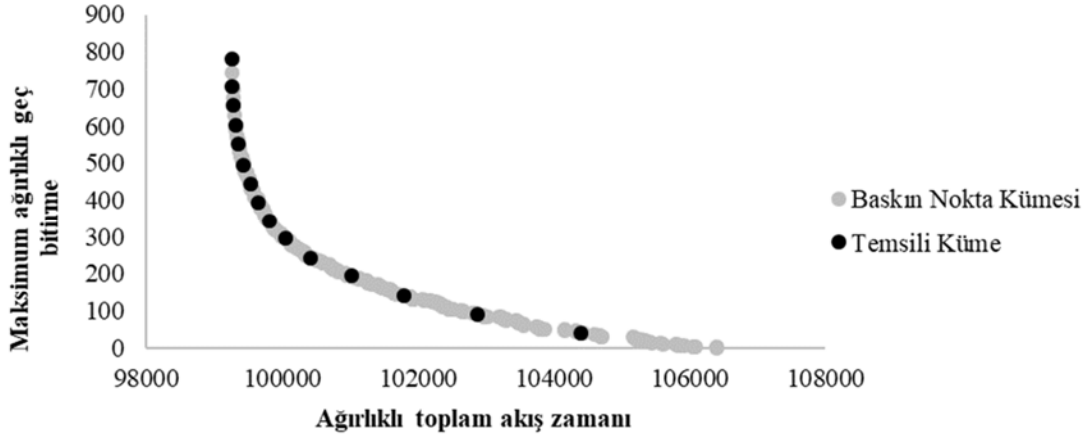
Tablo 8. Test problemlerinde algoritmanın performansı* (The performance of the algorithm on test instances)

İş Sayısı	α	Baskın Nokta Sayısı ($ Z_T $)	Çözüm Süresi (saniye)
100	1	262,04	0,0077
	25	85,04	0,0023
	50	54,64	0,0013
	75	41,32	0,0012
	100	33,60	0,0009
200	1	882,88	0,0910
	25	222,04	0,0221
	50	135,96	0,0144
	75	100,92	0,0100
	100	80,28	0,0081
300	1	1636,32	0,3509
	25	333,64	0,0715
	50	200,44	0,0424
	75	146,24	0,0317
	100	115,24	0,0234
400	1	2925,04	0,9276
	25	519,56	0,1766
	50	306,36	0,1033
	75	221,24	0,0699
	100	172,60	0,0571
500	1	4075,40	2,0094
	25	672,84	0,3527
	50	391,00	0,2375
	75	280,60	0,1607
	100	219,40	0,1170
600	1	5542,44	3,8752
	25	826,00	0,6404
	50	474,20	0,3618
	75	337,92	0,2683
	100	261,56	0,1970
700	1	7225,60	6,9008
	25	1027,60	1,0392
	50	587,16	0,5667
	75	419,00	0,4328
	100	323,60	0,3332
800	1	8607,48	10,3142
	25	1178,20	1,5304
	50	670,80	0,8119
	75	475,60	0,5853
	100	368,04	0,4984
900	1	10353,28	15,9382
	25	1355,28	2,2274
	50	766,72	1,2477
	75	543,84	0,8930
	100	420,92	0,7133
1000	1	12436,36	24,0537
	25	1591,80	3,2368
	50	898,84	1,8249
	75	633,36	1,2757
	100	300,82	0,6545

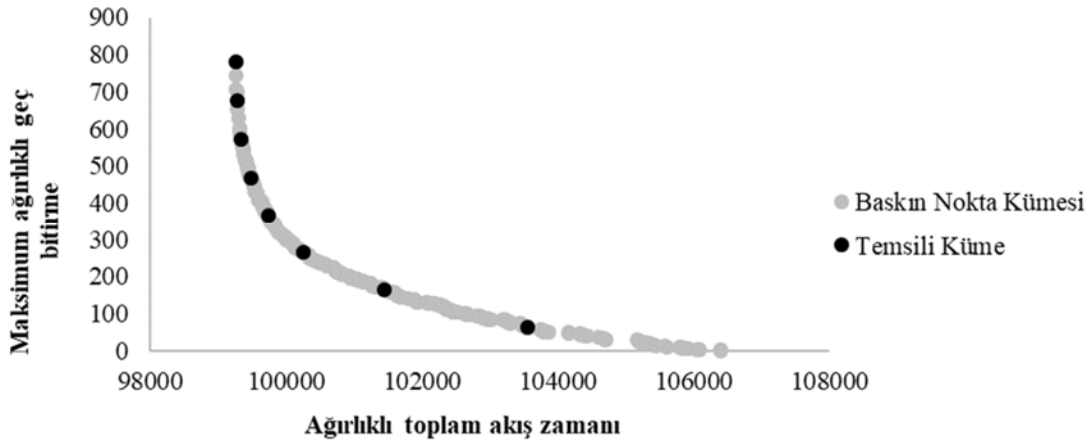
*50 farklı problem üzerinden ortalama değerler sunulmuştur. α değeri istenilen kapsama değerini göstermektedir. $\alpha = 1$ iken $Z_T = Z_B$.



Şekil 1. 100-işli tek makinalı çizelgeleme problemi için temsili küme ve baskın nokta kümesi
 $|Z_T| = 28$, $|Z_B| = 250$, $\alpha = 25$ (The nondominated set and representative set for 100-job single machine scheduling problem)



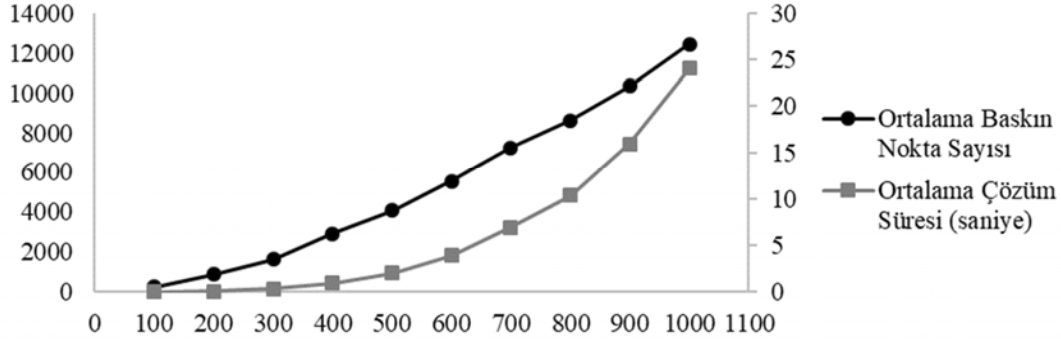
Şekil 2. 100-işli tek makinalı çizelgeleme problemi için temsili küme ve baskın nokta kümesi
 $|Z_T| = 15$, $|Z_B| = 250$, $\alpha = 50$ (The nondominated set and representative set for 100-job single machine scheduling problem)



Şekil 3. 100-işli tek makinalı çizelgeleme problemi için temsili küme ve baskın nokta kümesi
 $|Z_T| = 8$, $|Z_B| = 250$, $\alpha = 100$ (The nondominated set and representative set for 100-job single machine scheduling problem)

dolayısıyla çözüm süresi artmaktadır. Öyle ki, tüm parametre değerleri tamsayı seçildiğinden $\alpha = 1$ durumunda tüm baskın noktalar ve onlara karşılık gelen çizelgeleme çözümleri üretilmektedir. Deneylelerimizde yer alan

problemlerin baskın nokta sayısı [37, 27426] arasında yer almaktadır. Tablo 8'de yer alan sonuçlara göre, deney yapılan 100-işli problemlerin ortalama baskın nokta sayısı 262,04 iken 1000-işli problemlerde ortalama baskın nokta



Şekil 4. Farklı büyüklükte 50 tek makineli çizelgeleme problemi için ortalama baskın nokta sayıları ve ortalama çözüm süreleri (Average number of nondominated points and solution times for different-sized 50 single machine scheduling problems)

sayısı 12436,36 değerine yükselmektedir. Ek olarak, Şekil 4'te gösterildiği gibi çözüm süreleri problem büyüklüğü arttıkça hızla artmaktadır. Her ne kadar bu örneklerde çözüm süreleri makul seviyede kalsa da, karar vericilerin bu çözümleri değerlendirme zorluğu düşünüldüğünde tüm baskın noktaları bulmak yerine temsili bir altküme bulmak daha pratik bir yaklaşımdır. Örneğin, deneylerimizde yer alan ortalama baskın nokta sayısı 12436,36 olan 1000-ışli problemlerde $\alpha = 100$ değeri ile TÇBA uygulandığında, ortalama 0,6545 saniyede tüm bu baskın noktaları iyi temsil eden ortalama 300,82 çözüm üretilmektedir.

5. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Giderek karmaşıklaşan günümüz problemlerinde karar vericiler birden fazla birbiri ile çelişen kriterleri dikkate alarak çözüm bulmayı amaçlar. Ancak çok kriterli bu problemlerde, problem büyüklüğü arttıkça etkin çözüm sayısının artması karar vericinin bu çözümler içinden seçim yapmasını da zorlaştırmaktadır. Bu nedenle belirli baskın nokta kümesini istenen kalite ölçüsüne göre iyi temsil eden altküme bulmak pratik bir yaklaşım olarak kullanılabilir.

Bu çalışmada, mevcut çalışmalardan farklı olarak istenilen kapsama ölçüsüne göre etkin çizelgeleme altkümeleri bulmayı garantileyen bir yaklaşım önerilmiştir. Ayrıca, bu yaklaşımda, çizelgelenecek işlerin sadece işlem süresi ve teslim tarihi değil, aynı zamanda ağırlıkları da dikkate alınmaktadır. Bu ağırlıklar, karar verici tarafından belirlenebileceği gibi; işlerin kullandığı kaynak, risk, gecikme cezası gibi diğer faktörlere göre de belirlenebilir. Bu ağırlıklara göre hesaplanan toplam ağırlıklı akış zamanı ve ağırlıklı maksimum geç bitirme amaçlarını birlikte dikkate alan bir yaklaşım geliştirilmiştir.

Geliştirilen yaklaşım, TÇBA, farklı büyüklükteki ve farklı özelliklerdeki 500 problem üzerinde 5 farklı kalite seviyesi ile uygulanmıştır. Yapılan bu kapsamlı deneyler, tüm baskın nokta kümesini bulmak yerine altküme üretmenin hem çözüm süresi hem karar vericinin değerlendirme süreci açısından TÇBA'nın pratik bir yaklaşım olduğu göstermektedir. İleriki çalışmalarda, bu algoritmayı

geliştirerek karar vericinin de sürece dahil edildiği etkileşimli algoritmalar tasarlamak mümkündür. Bu sayede, gerçek hayat uygulamalarında istenilen kalite seviyesinin algoritma boyunca karar verici ile etkileşim kurarak belirlenmesi gerekirse algoritma sırasında güncellenmesi düşünülebilir.

KAYNAKLAR (REFERENCES)

1. Ehrgott, M. ve Gandibleux, X., A survey and annotated bibliography of multiobjective combinatorial optimization, OR Spektrum, 22, 425-460, 2000.
2. Dileepan, P. ve Sen, T., Bicriterion static scheduling research for a single machine, Omega, 16 (1), 53-59, 1988.
3. Fry, T.D., Armstrong, R.D. ve Lewis, H., A framework for single machine multiple objective sequencing research, Omega, 17 (6), 595-607, 1989.
4. Nagar, A., Haddock, J. ve Heragu, S., Multiple and bicriteria scheduling: A literature survey, European journal of operational research, 81 (1), 88-104, 1995.
5. Eren T., ve Güner E., A literature survey for multicriteria scheduling problems on single and parallel machines, Journal of the Faculty of Engineering and Architecture of Gazi University, 17 (4), 37-69, 2002.
6. Yen, B., ve Wan, G., Single machine bicriteria scheduling: A survey, International Journal of Industrial Engineering: Theory Applications and Practice, 10, 222-231, 2003.
7. T'kindt V ve Billaut J-C, Multicriteria scheduling: theory, models and algorithms. Springer, Berlin, 2002.
8. Hoogeveen, H., Multicriteria scheduling, European Journal of Operational Research, 167 (3), 592-623, 2005.
9. Van Wassenhove, L. N., ve Gelders, F., Solving A Bicriterion Scheduling Problem, European Journal of Operational Research, 4 (1), 42-48, 1980.
10. Smith, W. E., Varios Optimizers for Single-Stage Production, Naval Research Logistics Quarterly, 3, 59-66, 1956.
11. Sen, T. ve Gupta, S.K., A branch and bound procedure to solve a bicriterion scheduling problem, IIE Transactions, 15, 84-88, 1983.

12. Kondakçı, S.K., Azizoğlu, M. ve Köksalan, M., Note: Bicriteria scheduling for minimizing flow time and maximum tardiness, *Naval Research Logistics*, 43 (6), 929-936, 1996.
13. Köksalan, M.M., A heuristic approach to bicriteria scheduling, *Naval Research Logistics*, 46 (7), 777-789, 1999.
14. Lin, K.S., Hybrid algorithm for sequencing with bicriteria, *Journal of Optimization Theory and Applications*, 39 (1), 105-124, 1983.
15. Nelson, R.T., Sarin, R.K. ve Daniels, R.L., Scheduling with multiple performance measures: the one-machine case, *Management Science*, 32 (4), 464-479, 1986.
16. Chen, C.L. ve Bulfin, R.L., Scheduling a single machine to minimize two criteria: Maximum tardiness and number of tardy jobs, *IIE transactions*, 26 (5), 76-84, 1994.
17. Erenay, F.S., Sabuncuoglu, I., Toptal, A. ve Tiwari, M.K. New solution methods for single machine bicriteria scheduling problem: Minimization of average flowtime and number of tardy jobs, *European Journal of Operational Research*, 201 (1), 89-98, 2010.
18. Güner, E., Erol, S. ve Tani, K., One machine scheduling to minimize the maximum earliness with minimum number of tardy jobs, *International Journal of Production Economics*, 55 (2), 213-219, 1998.
19. Karasakal, E.K. ve Köksalan, M., A simulated annealing approach to bicriteria scheduling problems on a single machine, *Journal of Heuristics*, 6 (3), 311-327, 2000.
20. Köksalan, M. ve Keha, A.B., Using genetic algorithms for single-machine bicriteria scheduling problems. *European Journal of Operational Research*, 145 (3), 543-556, 2003.
21. Köksalan, M., Azizoğlu, M. ve Kondakçı, S.K., Minimizing flowtime and maximum earliness on a single machine, *IIE transactions*, 30 (2), 192-200, 1998.
22. Azizoğlu, M., Kondakçı, S. ve Köksalan, M., Single machine scheduling with maximum earliness and number tardy, *Computers and Industrial Engineering*, 45 (2), 257-268, 2003.
23. Shabtay, D. ve Kaspi, M., Minimizing the total weighted flow time in a single machine with controllable processing times. *Computers & Operations Research*, 31 (13), 2279-2289, 2004.
24. Wan, G. ve Yen, B.P.C., Single machine scheduling to minimize total weighted earliness subject to minimal number of tardy jobs, *European Journal of Operational Research*, 195 (1), 89-97, 2009.
25. Huo, Y., Leung, J.Y.T. ve Zhao, H., Bi-criteria scheduling problems: Number of tardy jobs and maximum weighted tardiness. *European Journal of Operational Research*, 177 (1), 116-134, 2007.
26. He, C., Lin, H. ve Wang, X., Single machine bicriteria scheduling with equal-length jobs to minimize total weighted completion time and maximum cost. *4OR*, 12 (1), 87-93, 2014.
27. Steiner, G. ve Stephenson, P., Pareto optima for total weighted completion time and maximum lateness on a single machine, *Discrete Applied Mathematics*, 155 (17), 2341-2354, 2007.
28. Sayin, S., Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming, *Mathematical Programming, Ser. A* 87, 543-560, 2000.
29. Wu, J. ve Azarm, S., Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set," *Journal of Mechanical Design, Transactions of the ASME*, 123 (1), 18-25, 2001.
30. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., ve Grunert da Fonseca, V. Performance Assessment of Multiobjective Optimizers: An Analysis and Review, *IEEE Transactions on Evolutionary Computation*, 7 (2), 117-132, 2003.
31. Faulkenberg, S.L. ve Wiecek, M.M., On the quality of discrete representations in multiple objective programming. *Optimization and Engineering*, 11 (3), 423-440, 2010.
32. Sylva, J., Crema, A., A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs," *European Journal of Operational Research*, 180 (3), 1011-1027, 2007.
33. Masin, M. ve Bukchin, Y., Diversity Maximization Approach for Multiobjective Optimization," *Operations Research*, 56 (2), 411-424, 2008.
34. Ceyhan, G., Köksalan, M. ve Lokman, B., Finding a representative nondominated set for multi-objective mixed integer programs, *European Journal of Operational Research*, 272 (1), 61-77, 2019.
35. Baker, K.R. ve Martin, J.B., An experimental comparison of solution algorithms for the single machine tardiness problem, *Naval Research Logistics Quarterly*, 21, 187-200, 1974.
36. Rinnooy Kan, A.H.G., Lageweg, B.J. ve Lenstra, J.K., Minimizing total costs in one machine scheduling, *Operations Research* 23, 908-927, 1975.
37. Ow, P.S. ve Morton, T.E., The single machine early/tardy problem, *Management Science* 35 (2), 177-191, 1992.