



A hybrid heuristic solution based on simulated annealing algorithm for energy efficient single machine scheduling problem with sequence dependent setup times

Gülçin Bektur*^{ID}

Industrial Engineering, Iskenderun Technical University, Hatay, 31200, Turkey

Highlights:

- Energy efficient single machine scheduling problem
- Multi- objective heuristic algorithm
- Sequence dependent setup times

Keywords:

- Energy efficient single machine scheduling
- ϵ - Constraint method
- Mixed integer linear programming model
- Multi- objective heuristic algorithm
- NSGA II algorithm

Article Info:

Research Article
Received: 29.07.2019
Accepted: 08.09.2020

DOI:

10.17341/gazimmfd.598082

Correspondence:

Author: Gülçin Bektur
e-mail:
gulcin.bektur@iste.edu.tr
phone: +90 326 613 5600

Graphical/Tabular Abstract

In this study, energy efficient single machine scheduling problem with sequence dependent setup times is considered. Speed scaling mechanism is taken in to account. Objective functions of the problem are minimization of the total number of tardy jobs and total energy consumption. A multi- objective heuristic algorithm is proposed for the problem. The proposed heuristic algorithm is compared with NSGA II and ϵ -constraint method. The Pareto front obtained by NSGA II and proposed heuristic algorithm is given in Figure A. The proposed heuristic algorithm outperforms NSGA II significantly.

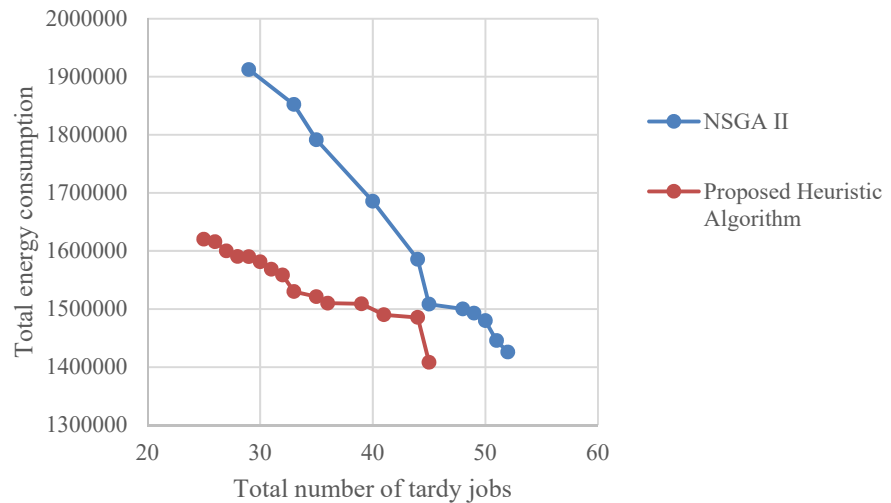


Figure A. Pareto front of the problem

Purpose: The aim of the study is to propose a successful heuristic algorithm for large scale problems due to the problem is addressed is NP- hard.

Theory and Methods:

A MILP (Mixed integer linear programming) model is proposed for the problem. Due to the NP- hard nature of the problem a multi- objective heuristic algorithm is proposed for the problem. The results of the algorithm is compared with state- of - the - art NSGA II algorithm.

Results:

Experimental studies shows the proposed heuristic algorithm found complete Pareto front for problems with 10 jobs. For test problems with more than 10 jobs proposed heuristic algorithm outperforms NSGA II significantly.

Conclusion:

By means of this study a different and successful algorithm is proposed for the problem. In future studies, an exact algorithm or matheuristic algorithms can be proposed for the problem.



Enerji etkin ve sıra bağımlı hazırlık süreli tek makine çizelgeleme problemi için tavlama benzetimi algoritması tabanlı hibrit sezgisel çözüm önerisi

Gülçin Bektur*^{ID}

İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Endüstri Mühendisliği Bölümü, 31200 İskenderun Hatay, Türkiye

Ö N E Ç İ K A N L A R

- Enerji etkin tek makine çizelgeleme problemi
- Çok amaçlı sezgisel algoritma
- Sıra bağımlı hazırlık süreleri

Makale Bilgileri

Araştırma Makalesi
Geliş: 29.07.2019
Kabul: 08.09.2020

DOI:

10.17341/gazimmfd.598082

Anahtar Kelimeler:

Enerji etkin tek makine çizelgeleme, ϵ -kısıt yöntemi, çok amaçlı sezgisel algoritma, sıra bağımlı hazırlık süreleri, çok amaçlı matematiksel model

ÖZET

Artan karbondioksit emisyonu ve enerji tüketimi, enerji etkin çizelgeleme problemlerine ilgiyi arttırmıştır. Bu çalışmada çok amaçlı enerji etkin tek makine çizelgeleme probleminde enerji tasarrufunun sağlanması amacıyla hız ayarlama metodu ele alınmıştır. Literatürde hız ayarlama metodunda, makina yüksek hızda çalıştırıldığında enerji tüketim oranı artmakta fakat işler daha kısa sürede tamamlanmaktadır. Makine düşük hızda çalıştırıldığında ise enerji tüketim oranı azalmakta fakat işlerin tamamlanma zamanı da uzamaktadır. Çalışmada, amaç fonksiyonları toplam tüketilen enerji miktarının ve geciken iş sayısının enküçüklenmesidir. Problem plastik parça üreten bir üretim işletmesindeki çizelgeleme problemi üzerinden tanımlanmıştır. İşler sıra bağımlı hazırlık süresine sahip olup, bir işin hazırlık süresi kendisinden önce tamamlanan işe bağlıdır. Probleme özgü bir matematiksel model önerilmiştir. Problemin çok amaçlı çözümünde ϵ -kısıt yöntemi kullanılmıştır. İlgili yöntemin kullanımıyla küçük boyutlu problemler için tüm Pareto etkin çözümler elde edilmektedir. Fakat problemin polinom zamanda çözülememesi sonucu büyük boyutlu problemlerin çözümü için çok amaçlı sezgisel bir algoritma önerilmiştir. Önerilen algoritmanın başarısı Non-dominated Sorting Genetic Algorithm II ile karşılaştırılarak gösterilmiştir.

A hybrid heuristic solution based on simulated annealing algorithm for energy efficient single machine scheduling problem with sequence dependent setup times

H I G H L I G H T S

- Energy efficient single machine scheduling problem
- Multi- objective heuristic algorithm
- Sequence dependent setup times

Article Info

Research Article
Received: 29.07.2019
Accepted: 08.09.2020

DOI:

10.17341/gazimmfd.598082

Keywords:

Energy efficient single machine scheduling, ϵ - constraint method, multi- objective heuristic algorithm, sequence dependent setup times, multi- objective mathematical model

ABSTRACT

Increased carbon dioxide emissions and energy consumption increase interest in energy efficient scheduling problems. In this study, speed scaling method is discussed in order to provide energy saving in multi-objective energy efficient single machine scheduling problem. In the literature with the speed scaling method, the energy consumption rate increases when the machine is operated at high speed, but the job is completed in a shorter time. When the machine is operate at low speed, the energy consumption rate decreases but the completion time of the jobs are prolonged. In the study, the objective functions are minimization of the total amount of energy consumption and the number of tardy jobs. The problem is defined by the scheduling problem in a manufacturing firm that produces plastic parts. Jobs have a sequence dependent setup times, and the setup time of a job depends on the job completed before it. A mathematical model is proposed for the problem. The ϵ -constraint method is used for the solution of the problem. All Pareto efficient solutions for small size problems are obtained by using the proposed heuristic method. However, as a result of the problem not being solved in polynomial time, a multi-objective heuristic algorithm is proposed for solving large-scale problems. The success of the proposed algorithm is shown by comparing with Non-dominated Sorting Genetic Algorithm II.

*Sorumlu Yazar/Corresponding Author: gulcin.bektur@iste.edu.tr / Tel: +90 326 613 5600

1. GİRİŞ (INTRODUCTION)

Hayatın devamlılığı için enerji vazgeçilemez, kıt bir kaynaktır. Enerjinin etkin kullanımı ise oldukça önemlidir. Çünkü enerji üretimi ve tüketimi süresince büyük miktarlarda sera gazı atmosfere salınmaktadır. Son yıllarda etkin olmayan enerji tüketimi ve büyük oranda sera gazı salınımı Dünya çapında bir problem haline gelmiştir. Bu nedenle araştırmacılar enerji tasarrufunun sağlanması konularına yönelmişlerdir. Ayrıca Uluslararası enerji ajansına göre 2040 yılında küresel enerji talebi %37 oranında artış gösterecektir [1]. Sonuçta enerjinin etkin kullanımı giderek önem kazanan bir konudur. Dünya genelinde üretim endüstrisi mevcut enerjinin neredeyse yarısını kullanmaktadır. Üretim endüstrisi enerjinin büyük bir kısmını kullandığı için atmosfere sera gazı salınımında da etkisi oldukça fazla olmaktadır. Enerjinin etkin kullanımı ve atmosfere sera gazı salınımının kontrolü için üretim endüstrisinin dikkate alınması son derece önemlidir [2]. Üretim endüstrisinde ise pek çok çözümlenmiş çalışmada sadece üretim etkinliği ile ilgili amaçlar dikkate alınmaktadır. Örneğin işlerin gecikme miktarının, tamamlanma zamanının, geciken iş sayısının enküçüklenmesi vb. amaçlar ele alınmaktadır. Oysa enerji etkin çözümlenmiş problemlerinde üretim etkinliğinin yanında enerji tasarrufu sağlayacak stratejiler de dikkate alınmaktadır. Literatürde enerji- etkin çözümlenmiş problemlerinde ele alınan stratejiler dört sınıfta düşünülmektedir [2]. 1. Hız ayarlama stratejisi (Speed scaling strategy): Makinaların hızları üretim etkinliği mümkün olduğunca az etkilenecek şekilde azaltılmaktadır. Burada üretim etkinliği ile ilgili amaçlar ve enerji ile ilgili amaçlar çalışmaktadır. Çünkü makinaların hızları azaltıldığında enerji tüketim oranı azalmakta, işlerin tamamlanma süreleri ise uzamaktadır. Amaç ise makinaların işlem hızının, tüketilen enerji miktarı ve üretim etkinliği bir arada dikkate alınarak belirlenmesidir.

2. Güç kapatma stratejisi (Power down strategy): Makinalar ardışık iki işlem arasında boş duruma geldiğinde kapatılmaktadır. Fakat makinaların tekrar çalışır duruma getirmek de belirli bir maliyete neden olmaktadır. Burada amaç makine boş duruma geldiğinde kapatılması ve sıradaki iş işlem için hazır olunca tekrar çalışır duruma getirilmesi sonucu kullanılan enerji mi ya da makinanın boş duruma çalışması sonucu kullanılan enerjinin mi daha tasarruflu olduğunun karar verilmesidir. Çünkü bazen ardışık iki iş arasında boş beklemler çok uzunsa makinanın kapatılıp, sıradaki iş işlem için hazır duruma gelince açılması daha mantıklıdır [3]. 3. Dinamik elektrik fiyatları (Dynamic electricity prices): Bu strateji ile elektriğin yüksek fiyatlandırıldığı periyotlarda makinaların mümkün olduğunca az çalıştırılması planlanmaktadır. Amaç toplam elektrik maliyetlerinin enküçüklenmesidir.

4. Önceden belirlenmiş toplam enerji tüketimi kısıtı (Pre-determined total energy consumption constraint): Çözümlenmiş periyodu boyunca önceden belirlenmiş toplam

enerji tüketimi miktarı aşılmayacak şekilde bir çözümlenmektedir. Enerji etkin tek makina çözümlenmiş literatürü incelendiğinde pek çok çalışmada güç kapatma stratejisinin ve dinamik elektrik fiyatlarının dikkate alındığı görülmüştür. Güç kapatma stratejisini dikkate alan çalışmalardan Mouzon ve Yıldırım [4], toplam gecikme ve toplam enerji tüketiminin enküçüklenmesinde sezgisel bir algoritma önermişlerdir. Mouzon vd. [5], ise atama kuralları ve matematiksel model önermişlerdir. Yıldırım ve Mouzon [6], toplam tamamlanma zamanı ve enerji tüketiminin enküçüklenmesinde çok amaçlı genetik algoritma (GA) önermişlerdir. Gong vd. [7], toplam enerji maliyetinin enküçüklenmesi için matematiksel model önermişlerdir. Che vd. [8], en büyük gecikme ve toplam enerji tüketiminin enküçüklenmesinde ϵ - kısıt yöntemini kullanmışlardır. İncelenen çalışmada, enerji etkin strateji olarak güç kapatma stratejisi benimsenmiştir. Yapılan çalışmada ise enerji etkin strateji makinalarda hız ayarlama stratejisidir. Amaç fonksiyonları açısından da çalışmalar farklıdır. İlgili çalışmada amaç fonksiyonları toplam enerji tüketim miktarının ve enbüyük gecikmenin enküçüklenmesidir. Yapılan çalışmada ise tüketilen toplam enerji miktarı ve toplam geciken iş sayısı enküçüklenmiştir. Ayrıca [8] numaralı kaynaktan sıra bağımlı hazırlık süreleri de dikkate alınmamıştır. Bu anlamda ele alınan problemler birbirlerinden farklıdır. Tarakçı vd. [9], toplam enerji tüketiminin enküçüklenmesi amaç fonksiyonunu dikkate almışlar ve problemin çözümünde GA önermişlerdir. Dinamik elektrik fiyatlarını dikkate alan çalışmalarda ise amaç fonksiyonu pek çok çalışmada toplam enerji maliyetinin enküçüklenmesi olup, tek amaçlıdır. Toplam enerji maliyetinin enküçüklenmesi çalışmalardan Shrouf vd. [10], GA önermişlerdir. Tsao vd. [11], toplam enerji maliyetinin enküçüklenmesi için bulanık GA önermişlerdir. Zhou vd. [12], toplam enerji maliyeti ve en büyük tamamlanma zamanının enküçüklenmesi için sezgisel algoritma önermişlerdir. Mikhaylidi vd. [13], işlerin bölünebilir olduğu durum için dinamik programlamayı kullanmışlardır. Aghelinejad vd. [14] ve Gong vd. [15], probleme özgü matematiksel model önermişlerdir. Gong vd. [16], problemi stokastik yapıda ele almışlar ve matematiksel model önermişlerdir. Aghelinejad vd. [17], işlerin bölünebilir olduğunu kabul etmişler ve toplam enerji maliyetinin enküçüklenmesi problemler için karmaşıklık analizi yapmışlardır. Gong vd. [18], toplam enerji maliyetinin yanında toplam işgücü maliyetini de dikkate almışlar ve sezgisel bir algoritma önermişlerdir. Lee vd. [19], erkenlik ve geçlik için toplam ceza maliyetini ve toplam enerji maliyetini enküçükleyen çok amaçlı sezgisel bir algoritma önermişlerdir. Aghelinejad vd. [20] ve Zhang vd. [21], sezgisel algoritma önermişlerdir. Rubaiee vd. [22], toplam gecikme ve toplam enerji maliyetini dikkate almışlar ve problemin çözümünde çok amaçlı GA önermişlerdir. Önceden belirlenmiş enerji tüketim kısıtı aşılmayacak şekilde tek makina çözümlenmiş problemini inceleyen çalışmalardan, Artigues vd. [23], iki aşamalı kısıt programlama yaklaşımı önermişlerdir. Nattaf vd. [24], ise hibrit dal-sınır algoritması önermişlerdir. Ayrıca

çizelgelemede kaynak kullanımını dikkate almışlardır. Liao vd. [25], toplam ağırlıklı tamamlanma zamanı ve toplam ağırlıklı gecikme amaç fonksiyonlarını dikkate almışlardır. Problemin çözümünde çok amaçlı parçacık sürüsü algoritması önermişlerdir. Chen vd. [26], makinaların fiziksel koşullarını dikkate almışlar ve toplam gecikme ve enerji tüketim maliyetinin enküçüklendiği problemin çözümü için Karınca Kolonisi algoritmasını önermişlerdir.

Hız ayarlama stratejisini dikkate alan çalışmalar diğer çalışmalara göre daha azdır. Hız ayarlama stratejisini dikkate alan tek makina çizelgeleme çalışmalarından Albers ve Fujiwara [27], toplam akış zamanı ve toplam enerji maliyeti enküçükleyen bir algoritma önermişlerdir. Antoniadis ve Huang [28], hız ayarlama stratejisini dikkate alan toplam enerji tüketiminin enküçüklendiği tek makina çizelgeleme probleminin NP- zor olduğunu göstermişlerdir. Problemi tek amaçlı olarak ele almışlardır. Antoniadis vd. [29], toplam enerji tüketimi amaç fonksiyonunun dikkate alındığı problem için yeni bir algoritma geliştirmişlerdir. Fang vd. [30], toplam enerji maliyetinin enküçüklenmesi amaç fonksiyonunu incelemişlerdir. Hız ayarlama stratejisini ve dinamik elektrik fiyatlarını dikkate almışlardır. Problemin çözümü için bir algoritma geliştirmişlerdir. Tasgetiren vd. [31], tek makina çizelgeleme probleminde hız ayarlama stratejisini dikkate almışlardır. Yapılan çalışmada [31] numaralı çalışmaya göre incelenen problem ve çözüm yöntemi farklıdır. İlgili çalışmada amaç fonksiyonu toplam enerji tüketimi ve toplam ağırlıklı gecikmenin enküçüklenmesidir. Ele alınan problemde ise toplam geciken iş sayısı ve toplam enerji tüketimi enküçüklenmiştir. İncelenen çalışmada iki farklı sezgisel algoritma önerilip, birbiriyle karşılaştırılmıştır. Matematiksel modelin kullanımıyla küçük boyutlu problemler için Pareto optimal yüzey elde edilmemiştir. Bu çalışmada farklı olarak küçük boyutlu problemler için ϵ - kısıt yönteminin kullanımıyla Pareto optimal yüzey elde edilmiştir. Böylece küçük boyutlu problemlerde önerilen sezgisel algoritmanın sonuçları Pareto optimal yüzey ile karşılaştırılabilir. Ayrıca yapılan çalışmada problemin çözümünde farklı bir sezgisel algoritma önerilip, önerilen algoritmanın başarısı literatürdeki NSGA II (Non-dominated Sorting Genetic Algorithm II) algoritmasıyla karşılaştırılmıştır. [31] numaralı çalışmada en çok 60 iş için sezgisel algoritma çalıştırılmıştır. Yapılan çalışmada ise problem boyutu daha büyük alınmıştır. Çalışmaların benzer özellikleri ise sıra bağımlı hazırlık sürelerinin ve makinalarda hız ayarlama stratejisinin dikkate alınmasıdır. Toplam geciken iş sayısının enküçüklenmesi amaç fonksiyonu pek çok çalışmada dikkate alınan önemli bir amaç fonksiyonu olmasına rağmen enerji etkin çizelgeleme problemlerinde daha önce ele alınmamıştır. Toplam geciken iş sayısının dikkate alındığı bazı çalışmalar verilmiştir. Seo vd. [32], toplam beklenen geciken iş sayısının en küçükleme amaç fonksiyonunu dikkate almışlardır. Stokastik yapıdaki tek makine çizelgeleme problemi için matematiksel model önerilmiştir. Chen [33], tek makine çizelgeleme probleminde periyodik bakımı dikkate almıştır. Amaç fonksiyonu toplam geciken iş sayısının enküçüklenmesi olup, dal- sınırlı algoritması

önerilmiştir. Cyril ve Samia [34], toplam geciken iş sayısının enküçüklendiği tek makine çizelgeleme problemi için alt sınırın ve üst sınırın dikkate alındığı bir matematiksel model önermişlerdir. Liu ve Liao [35], toplam erkenlik ve geçlik ile toplam enerji tüketiminin enküçüklendiği tek makine çizelgeleme problemini incelemişlerdir. Aydılek vd. [36], tek makine çizelgeleme probleminde stokastik işlem sürelerini dikkate almıştır. Amaç fonksiyonunun toplam geciken iş sayısının enküçüklenmesi olan problemin çözümünde farklı bir algoritma geliştirmiştir. Bu çalışmada enerji etkin tek makine çizelgeleme probleminde hız ayarlama stratejisi dikkate alınmıştır. Amaç fonksiyonları toplam enerji tüketiminin ve geciken iş sayısının enküçüklenmesidir. Yüksek makine hızı işlerin hızlı bir şekilde tamamlanmasını sağlamakta fakat yüksek enerji tüketim oranına neden olmaktadır. Düşük makine hızı ise düşük enerji tüketim oranı sağlamakta fakat işlerin tamamlanma zamanını uzatmaktadır. Sonuçta enerji ve üretim etkinliği amaçları birbirleriyle çelişen amaçlar olup problem çok amaçlı ele alınmalıdır. Problemden işler sıra bağımlı hazırlık sürelerine sahip olup, bir işin hazırlık süresi kendisinden önce tamamlanan işe bağlıdır. Probleme özgü bir matematiksel model önerilmiştir. Problemin çözümünde tüm Pareto çözümlerinin bulunmasını sağlayan ϵ -kısıt yaklaşımı kullanılmıştır. Büyük boyutlu problemlerin çözümü için çok amaçlı sezgisel algoritma önerilmiştir. Küçük boyutlu problemlerde önerilen sezgisel algoritmanın başarısı ϵ - kısıt yöntemiyle karşılaştırılmıştır. Fakat ele alınan problemin NP- zor yapısı gereği büyük boyutlu problemlerde matematiksel modelde ϵ - kısıt yöntemi kullanılarak eniyi çözüm garanti edilememektedir. Bu nedenle önerilen algoritmanın başarısı küçük boyutlu problemlerde ϵ - kısıt yöntemi ile karşılaştırılarak ve büyük boyutlu problemlerde literatürdeki NSGA II algoritmasıyla karşılaştırılarak gösterilmiştir. Literatür dikkate alındığında, enerji etkin tek makina çizelgeleme problemlerinde hız ayarlama stratejisini dikkate alan çalışmalar az sayıdadır. Bu çalışmalardan sıra bağımlı hazırlık sürelerini dikkate alan sadece bir çalışmaya rastlanmıştır. Bu çalışmanın literatürden farkı, enerji etkin tek makina çizelgeleme problemlerinde toplam geciken iş sayısının ilk defa ele alınmasıdır. Ele alınan problem gerçek bir çizelgeleme probleminden tanımlanmıştır. Ayrıca önerilen ϵ - kısıt yönteminin kullanımıyla küçük boyutlu problemler için tüm Pareto etkin çözümler bulunmuştur. Büyük boyutlu problemler için literatürden farklı çok amaçlı sezgisel bir algoritma önerilmiştir. Bu algoritma literatürdeki NSGA II algoritmasıyla karşılaştırılmıştır. Çalışmanın ilk bölümü giriş bölümüdür. İkinci bölümde kullanılan matematiksel model ve ϵ -kısıt yaklaşımı anlatılmıştır. Üçüncü bölümde önerilen sezgisel algoritmaya yer verilmiştir. Dördüncü bölümde önerilen algoritmanın başarısı NSGA II ile karşılaştırılarak gösterilmiştir. Son bölüm sonuç bölümüdür.

2. PROBLEMİN TANIMLANMASI (DESCRIPTION OF THE PROBLEM)

Bu çalışmada ele alınan problem plastik enjeksiyon makinasının çizelgelemesi üzerinden tanımlanmıştır. Ele

alınan problem plastik parça üretiminin yapıldığı ve enjeksiyon makinasının kullanıldığı bir işletme dikkate alınarak belirlenmiştir. Firmada bir uygulama yapılmamış olup, sadece problemin tamamen gerçek hayat problemini yansıtmayı amacıyla firma dikkate alınmıştır. Bunun nedenlerinden biri de test problemlerinin firmanın gerçek verilerine göre daha büyük boyutlu olmasıdır. Bir başka deyişle firmadan alınacak veri daha küçük boyutlu olduğundan uygulama makaleye eklenmemiştir. Plastik parçalar plastik enjeksiyon makinası ile üretilmektedir. Plastik parça üreten firma farklı renk, boyut ve özelliklere sahip siparişler almaktadır. Üretimin gerçekleştirilebilmesi için alınan siparişe uygun kalıp enjeksiyon makinasına bağlanmalı, hammaddenin koyulduğu hazne temizlenmeli ve siparişe uygun hammadde hazneye alınmalıdır. Tüm bu işlemler hazırlık aşamasını oluşturmaktadır. Hazırlık süreleri sıra bağımlıdır. Birbirine benzer parçalar ardışık üretildiğinde hazırlık süresi farklı parçaların ardışık üretimine göre kısalmaktadır. Örneğin siyah bir parçanın üretiminden sonra beyaz parçanın üretimi olacaksa, kazanların çok iyi temizlenmesi gerekmektedir ve hazırlık süresi uzayabilmektedir. İncelenen enjeksiyon makinasının hızı ayarlanabilir özelliktedir. Makine hızlı çalıştırıldığında işlem süresi kısalmakta fakat enerji tüketim oranı ise artmaktadır. Makinanın çalışma aşamasında tüketilen enerji oranı dijital olarak görünmektedir. Firma geciken iş sayısının ve toplam enerji tüketiminin en küçüklemesini istemektedir. Bir iş işlenmeye başladıktan sonra makinanın hızı değiştirilememektedir. j işinin tamamlanması için gereken iş yükü L_j olarak gösterilmektedir. İşin işlem zamanı ise iş yükünün makine hızına bölünmesiyle bulunmaktadır. Bir işin tamamlanma zamanı ise kendisinden önce çizelgelenen işin tamamlanma zamanına ilgili işin işlem zamanının ve sıra bağımlı hazırlık süresinin eklenmesiyle bulunmaktadır. Eğer j işi q hızı ile işleniyorsa, enerji tüketim oranı p_{jq} ile gösterilir. j işi için makine hızı arttırılırsa enerji tüketim oranı p_{jq} artışı göstermektedir. Başka bir deyişle, x hızı y hızından büyükse ($v_x > v_y$), ilgili iş için x hızında enerji tüketim oranı y hızında enerji tüketim oranından büyük ($p_{jx} > p_{jy}$) olmaktadır. Makine yüksek hızda çalıştırıldığında, işin işlem süresi L_j/v_q ya eşit olduğu için, kısalmaktadır. Fakat enerji tüketim oranı da artmaktadır. Toplam enerji tüketimi ise işlem süreleri ile işlerin enerji tüketim oranının çarpılmasıyla bulunmaktadır. İş bölünmelerine izin verilmemektedir. Probleme özgü çok amaçlı matematiksel model aşağıda verilmiştir.

2.1. Önerilen Matematiksel Model (Proposed Mathematical Model)

Matematiksel model önerilirken [2, 34] tarafından yapılan çalışmalar dikkate alınmıştır.

Kümeler ve indisler

- N : İş kümesi, $N = \{1, 2, \dots, n\}$
 S : Makine hızı kümesi, $S = \{1, 2, \dots, s\}$
 j, i : İşleri göstermek için kullanılan indis, j ve $i \in N$
 k : Sırayı göstermek için kullanılan indis, $k \in N$

q : Hızı göstermek için kullanılan indis, $q \in S$

Parametreler

- L_j : j işinin iş yükü
 s_{ij} : i işinden sonra gelen j işinin sıra bağımlı hazırlık süresi
 d_j : j işinin teslim zamanı
 v_q : q hızı seviyesinde işlem yapan makinanın hızı
 p_{jq} : j işinin q hızı seviyesinde enerji tüketim oranı
 M : Yeterince büyük bir sayı

Karar Değişkenleri

- E : Toplam enerji tüketimi,
 T_j : j işinin gecikme miktarı,
 C_j : j işinin tamamlanma zamanı
 $U_j = \begin{cases} 1; & \text{Eğer } j \text{ işi gecikiyorsa} \\ 0; & \text{d. d.} \end{cases}$
 $x_{jk} = \begin{cases} 1; & \text{Eğer } j \text{ işi } k. \text{ sırada işlem görüyorsa} \\ 0; & \text{d. d.} \end{cases}$
 $y_{jq} = \begin{cases} 1; & \text{Eğer } j \text{ işi } q \text{ hızı seviyesinde} \\ 0; & \text{işlem görüyorsa} \\ & \text{d. d.} \end{cases}$

Model

$$\text{Min } \sum_j U_j \quad (1)$$

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

$$\text{s.t.} \\ \sum_k x_{jk} = 1 \quad \forall j \quad (3)$$

$$\sum_j x_{jk} = 1 \quad \forall k \quad (4)$$

$$C_j \geq \frac{L_j}{v_q} y_{jq} \quad \forall j, q \quad (5)$$

$$C_j \geq C_i - M(2 - x_{i(k-1)} - x_{jk}) + s_{ij} + \frac{L_j}{v_q} y_{jq} \\ \forall j, i, q \text{ ve } k > 1, i \neq j \quad (6)$$

$$E = \sum_j \sum_q \frac{L_j}{v_q} y_{jq} p_{jq} \quad (7)$$

$$C_j - d_j \leq T_j \quad \forall j \quad (8)$$

$$T_j \leq M U_j \quad \forall j \quad (9)$$

$$\sum_q y_{jq} = 1 \quad \forall j \quad (10)$$

$$C_j, E, T_j \geq 0 \quad \forall j \quad (11)$$

$$x_{jk}, y_{jq}, U_j \in \{0, 1\} \quad \forall j, k \text{ ve } q \quad (12)$$

Kısıt 1 ve Kısıt 2 amaç fonksiyonları olup, sırasıyla toplam geciken iş sayısının ve toplam enerji tüketiminin en

küçüklenmesidir. Kısıt 3 her işin bir sıraya atanmasını sağlar. Kısıt 4 her sıraya bir işin atanmasını sağlar. Kısıt 5 ile işlerin tamamlanma zamanının işlem süresinden büyük olması sağlanır. Kısıt 6) işlerin tamamlanma zamanlarını hesaplar. İşin tamamlanma zamanı ise kendisinden önceki işin tamamlanma zamanına ilgili işin sıra bağımlı hazırlık süresi ile işlem süresinin eklenmesiyle bulunur. Kısıt 7 enerji tüketim miktarını hesaplar. Toplam enerji tüketimi ise işlem süreleri ile işlerin enerji tüketim oranının çarpılmasıyla bulunmaktadır. Kısıt 8 işlerin gecikme miktarını hesaplar. Gecikme miktarı işlerin tamamlanma zamanı ile teslim zamanının farkıdır. Kısıt 9 geciken işlerin bulunmasını sağlar. Eğer işin gecikme miktarı sıfırdan büyükse iş gecikiyor demektir ve U_j değeri 1' e eşittir. Aksi halde U_j değeri sıfıra eşittir. Kısıt 10 her işin bir hız seviyesinde işlenmesini sağlar. Kısıt 11 ve Kısıt 12 işaret kısıtlarıdır.

2.2. ε -Kısıt Yöntemi (ε - Constraint Method)

ε - kısıt yöntemi çok amaçlı optimizasyonda Pareto etkin yüzeyin bulunmasında kullanılmaktadır. Çalışmada, toplam enerji tüketimi (F_2) amaç fonksiyonu olarak alınmakta ve toplam geciken iş sayısı (F_1) ise bir kısıt olarak eklenmektedir. ε - kısıt yöntemi aşağıda açıklanmıştır ve aşağıdaki matematiksel model *Problem X* olarak isimlendirilmiştir:

$$\text{Min } F_2$$

s. t.

$$\text{Kısıt 3-Kısıt 12 ve } F_1 \leq \varepsilon \quad (13)$$

Kısıt 13 toplam geciken iş sayısının ε değerine eşit ya da küçük olmasını sağlar. Başlangıçta ε değeri n olarak alınmakta ve *Problem X* çözülmektedir. Elde edilen çözüm ve amaç fonksiyonları (F_1, F_2) kaydedilir. Problemin çözümü sonucu bulunan toplam geciken iş sayısı 1 birim azaltılır ve ε değeri bu değere eşit alınır. *Problem X* tekrar çözülür. Algoritma ilgili ε değerinde uygun çözüm bulamayana kadar çalışır. Tüm Pareto etkin yüzeyin bulunmasında kullanılan ε -kısıt yöntemi Algoritmik olarak aşağıda verilmiştir. Benzer bir yöntem Bootaki vd. [37], tarafından çok amaçlı kübik hücre oluşturma probleminde kullanılmıştır.

Procedure ε - kısıt yöntemi

Input: Problem parametreleri

Output: Etkin çözümler

$\varepsilon = n; \Delta \leftarrow 1; \text{check} = 1; k = 1;$

While ($\text{check} = 1$)

Problem X çözülür;

If mümkün çözüm elde edildiyse

Çözüm k . çözüm olarak

kaydedilir; $k = k + 1; \varepsilon = F_1 - \Delta;$

Else

$\text{check} = 0;$

End

End

Etkin çözümleri kaydet

Aşağıda örnek bir problem verilmiştir. Bu problem 1 numaralı test problemidir.

Örnek problem için sıra bağımlı hazırlık süreleri (s_{ij}), işlerin yükü (L_j), teslim tarihleri (d_j) ve enerji tüketim oranları (p_{jq}) Tablo 1'de verilmiştir. Makine hızı 0,75, 1, 1,25 ve 1,50 olarak alınmış olup, $v_1=0,75, \dots, v_4=1,50$ ' dir. İlk etkin çözüm (4;89917,33) olarak bulunmuştur. Bu çözümde toplamda 4 iş gecikmekte ve toplam enerji tüketimi de 89917.33' dir. Toplam geciken iş sayısı 1 birim azaltılıp ε değeri 3 olarak alınır ve *Problem X* çözülür. Elde edilen çözüm (3;97320,67)' dir. Toplam geciken iş sayısı 1 birim azaltılıp ε değeri 2 olarak alınır. Elde edilen çözüm (2; 124724,53)' dir. ε değeri 1 alınıp, *Problem X* çözüldüğünde 100000 sn' de uygun çözüm bulunamadığı için, algoritma sonlandırılmıştır. Belirlenen U_j değerinde elde edilen eniyi çözümler Tablo 2' de verilmiştir.

Tablo 2: Örnek problem için Pareto etkin çözümler (Pareto efficient solutions for the problem)

$\sum U_j$	E	CPU
4	89917,33	266,94
3	97320,67	34269,13
2	124724,53	83090,05

Tablo 1. Örnek problemin parametreleri (Parameters of the example problem)

j	S_{ij}										L_j	d_j	p_{j1}	p_{j2}	p_{j3}	p_{j4}
	1	2	3	4	5	6	7	8	9	10						
1	-	15	9	14	9	12	7	6	6	10	39	162	110	195	305	439
2	12	-	4	13	7	13	14	13	7	15	48	168	432	768	1200	1728
3	8	14	-	6	9	11	4	13	4	14	17	171	153	272	425	612
4	5	10	3	-	7	6	12	4	10	8	9	136	46	81	127	182
5	4	9	10	7	-	3	7	11	5	11	45	165	253	450	703	1013
6	14	8	10	14	15	-	11	12	7	7	47	166	238	423	661	952
7	13	10	11	6	11	11	-	9	5	14	35	164	354	630	984	1418
8	9	7	5	9	9	5	7	-	15	14	17	162	163	289	452	650
9	12	15	6	8	12	4	11	11	-	11	12	166	108	192	300	432
10	15	9	15	3	7	5	13	9	12	-	9	174	41	72	113	162

3. ÖNERİLEN SEZGİSEL ALGORİTMA (PROPOSED HEURISTIC ALGORITHM)

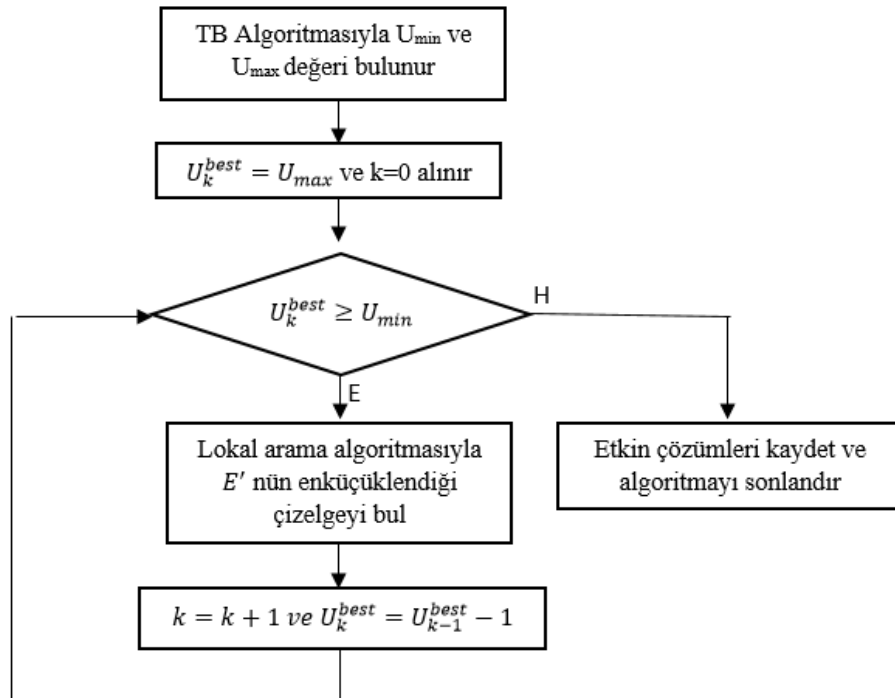
Önerilen sezgisel algoritmada öncelikle tüm işler için makine hızı en yüksek hız kabul edilip, Tavlama Benzetimi (TB) algoritmasıyla sıra bağımlı hazırlık süreli toplam geciken iş sayısının enküçüklendiği tek makine çizelgeleme problemi ($1|s_{ij}|\sum U_j$) çözülmekte ve çözüm sonucu incelenen problem için toplam geciken iş sayısı amaç fonksiyonunun alabileceği en küçük değer (U_{min}) belirlenmektedir. Toplam geciken iş sayısı amaç fonksiyonunun alacağı en büyük değerin (U_{max}) belirlenmesi için de makine hızı tüm işler için en düşük hız kabul edilmekte ve $1|s_{ij}|\sum U_j$ problemi TB algoritmasıyla çözülmektedir. U_k^{best} k. çözümde toplam geciken iş sayısını göstermektedir. Başlangıçta U_k^{best} değeri, U_{max} değerine eşit alınır. En çok U_k^{best} değerinde iş gecikecek ve toplam enerji tüketimini enküçükleyen çizelgenin bulunması için lokal arama algoritması kullanılmaktadır. Elde edilen çözüm kaydedilmekte ve U_k^{best} değeri bir birim azaltılarak lokal arama algoritması en çok ($U_k^{best}-1$) sayıda iş gecikecek ve toplam E enküçüklenecek şekilde tekrar çalıştırılmaktadır. Bu adım, U_k^{best} değeri U_{min} ' den küçük bir değere ulaşana kadar tekrar edilmekte ve etkin çözümler kaydedilmektedir. Eğer ilgili U_{min} değerine sahip çözüm bulunamazsa sonraki U_{min} değerine geçilir. Bu yöntemin kullanımıyla U_{min} ve U_{max} arasında toplam geciken iş sayısı olduğu durumda E nin enküçük değerinin bulunması amaçlanmaktadır. Sezgiselde kullanılan algoritmaların durdurma kriterlerini belirtmek gerekirse ana algoritmayı sonlandırma kriteri tavlama benzetimi algoritmasının bulunduğu minimum toplam geciken iş sayısına ulaşılmasıdır.

Çünkü bulunan bu değerden daha az sayıda işin geciktiği bir çizelge mümkün olmamaktadır. Lokal arama algoritmasının durdurma kriteri ise önceden belirlenmiş sayıda çözümün türetilmesidir. Tavlama benzetimi algoritması ise belirlenen son sıcaklığa ulaştığında durmaktadır. Algoritmayı sonlandırma kriterleri belirlenirken literatür ve yapılan testler dikkate alınmıştır. Algoritma durduktan sonra, etkin çözümler kaydedilmektedir. Önerilen sezgisel algoritmanın adımları Şekil 1'de verilmiştir.

Sezgisel algoritmanın yazımında kullanılan kısaltmalar verilmiştir. $T_{final(c)(0)}$: Final (mevcut)(başlangıç) sıcaklığı; $M_{2,max}$: Metod 1 sonucu elde edilen çözüme Metod 2' nin uygulanma sayısı; $maxtry$: Metod 2' nin kullanımı sonucu uygun olmayan çözüm sayısı; $V_{max(min)}$: en yüksek (en düşük) makine hızı; il : iterasyon sayacı; IT : Her bir sıcaklıktaki iterasyon sayısı; q : soğutma oranı; U_k^{best} : k. çözümde toplam geciken iş sayısı; $maxiter$: lokal arama algoritması için maksimum iterasyon sayısı; E_{best} : En iyi çözümde toplam enerji tüketim miktarı; U_{best} : En iyi çözümde toplam geciken iş sayısı; E_n : Komşu çözümde toplam enerji tüketim miktarı; S_n : Komşu çözüm; S_{best} : En iyi çözüm; $iter$: Lokal arama algoritması iterasyon sayacı

3.1. Tavlama Benzetimi Algoritması (Simulated Annealing Algorithm)

Tavlama benzetimi algoritması çoğu çizelgeleme probleminde başarılı bulunan sezgisel bir algoritmadır. Lokal eniyiden kaçmak için iyileşme sağlamayan çözümleri de belirli olasılıkla kabul eder. Algoritmanın sonlarına doğru iyileşme yapmayan çözümlerin kabul olasılığı azalır. Burada



Şekil 1. Sezgisel algoritmanın adımları (Steps of the heuristic algorithm)

amaç umut vaad eden bölgelerde aramayı yoğunlaştırmaktır. TB algoritması özellikle kapasite kısıtsız NP- zor problemlerin çözümünde çok başarılıdır. Ele alınan problem kapasite kısıtsız bir problem olduğu için TB algoritmasının mantığına uygundur. Bu nedenle TB algoritması tercih edilmiştir.

Çalışmada, TB algoritması makine hızı en düşük veya en yüksek hız kabul edildiğinde tek makine, sıra bağımlı hazırlık süreli ve toplam geciken iş sayısının en küçüklenmesi probleminin $(1|s_{ij}|\sum_j U_j)$ çözümü için kullanılmaktadır. Böylece toplam geciken iş sayısı amaç fonksiyonu için U_{max} ve U_{min} değerleri bulunmaktadır. TB algoritması başlangıç bir çözümle başlar. Bu çalışmada başlangıç çözüm EDD (Earliest due date first) kuralına göre oluşturulmaktadır. Komşu çözüm başlangıç çözümden türetilmektedir. Bu çalışmada rassal seçilmiş iki iş yer değiştirilerek komşu çözüm oluşturulmaktadır. Komşu çözümle mevcut çözüm amaç fonksiyonları arasındaki fark Δ ile gösterilir. Eğer Δ sıfırdan küçükse, komşu çözüm daha iyi bir çözüm demektir ve bu durumda mevcut çözüm olarak kabul edilir. TB algoritması yerel en iyiden kaçmak için iyileşme yapmayan çözümleri de belirli bir olasılıkla kabul eder. Bu amaçla rassal olarak türetilmiş sayı $exp(-\Delta/T_c)$ değerinden daha küçükse, iyileşme sağlamayan çözüm kabul edilir. T_c mevcut sıcaklığı göstermektedir. TB algoritmasında sıcaklık azaldıkça algoritmanın iyileşme yapmayan çözümleri kabul olasılığı düşmektedir. Önceden belirlenmiş iterasyon sayısı sağlanana kadar mevcut çözümden komşu çözüm türetilir ve Δ değerine göre çözüm mevcut çözüm olarak kabul ya da red edilir. Önceden belirlenmiş iterasyon sayısına ulaşılmca sıcaklık $T_c = T_c * q$ denklemi ile düşürülür. q soğutma oranını göstermektedir. Algoritma son sıcaklığa ulaşana kadar çalışmaktadır. TB algoritması aşağıda verilmiştir.

Procedure TB Algoritması

Input: $n, L_j, S_{ij}, d_j, p_{jq}, V_{\min(\max)}, T_0, T_{\text{final}}, q, IT, f$

Output: $U_{\min}, U_{\max};$

U_0 amaç fonksiyonuna sahip başlangıç çözüm (S_0) EDD kuralına göre oluşturulur;
 $S_{\text{best}}=S_c=S_0; U_{\text{best}}=U_c=U_0; T_c=T_0;$

If ($f==0$)

$V=V_{\max};$

Else

$V=V_{\min};$

End

While ($T_c > T_{\text{final}}$)

$il=0;$

While ($il < IT$)

U_n amaç fonksiyonuna sahip komşu çözüm (S_n) türetilir;

$il=il+1; \Delta=U_n-U_c;$

If ($\Delta < 0$) or ($\Delta \geq 0$ and random
 $(0-1) < exp(-\Delta/T_c)$)

$S_c=S_n; U_c=U_n;$

End

If ($U_c < U_{\text{best}}$)

$S_{\text{best}}=S_c; U_{\text{best}}=U_c;$

End

End

$T_c=T_c*q;$

End

If ($f==0$)

$U_{\min}=U_{\text{best}};$

Else

$U_{\max}=U_{\text{best}};$

End

3.2. Lokal Arama Algoritması (Local Search Algorithm)

Lokal arama algoritması NP- zor problemlerin çözümünde sıklıkla başvurulan bir algoritmadır. Lokal arama algoritmasının kullanımındaki amaç sezgiselin umut vaad eden bölgelerde yoğun arama yapmasını sağlamaktır. Lokal arama algoritması en çok U_k^{best} sayıda geciken iş sayısını sağlayan ve toplam enerji tüketiminin en küçüklendiği çözümlerin bulunması için kullanılmaktadır. Komşu çözümlerin oluşturulmasında farklı bir yöntem kullanılmıştır. Bu bölümde lokal arama algoritmasının adımları, çözümlerin gösterimi ve komşu çözümlerin nasıl oluşturulduğu anlatılmıştır.

3.2.1. Lokal arama algoritmasının adımları

(Steps of the local search algorithm)

Lokal arama algoritmasında önceden belirlenmiş sayıda komşu çözüm türetilmektedir. Türetilen komşu çözümlerde en çok U_k^{best} sayıda iş gecikecek şekilde komşu çözüm türetme yöntemi kullanılmaktadır. Türetilen komşu çözümlerden toplam enerji tüketiminin en küçük olduğu çözüm alınmaktadır. Bu çözüm eldeki en iyi çözümden daha iyi bir çözüme (toplam enerji tüketimi daha küçükse), en iyi çözüm güncellenmektedir. İterasyon sayısı bir artırılmakta ve algoritma önceden belirlenmiş iterasyon sayısına ulaşana kadar çalışmaktadır. Algoritmanın adımları aşağıda verilmiştir.

Procedure lokal arama algoritması

Input: Problem parametreleri, $M_{2\max}, \text{maxtry},$

$U_k^{\text{best}}, \text{maxiter}, IT$

Output: Etkin çözümler

$iter \leftarrow 1; E_{\text{best}} \leftarrow \infty;$

While ($iter \leq \text{maxiter}$)

$t \leftarrow 0;$

While ($t < IT$)

$(S_n^t, E_n^t) \leftarrow$ **Komsu çözüm türetme**

(problem data, $M_{2\max}, U_k^{\text{best}}, \text{maxtry}$);

$t \leftarrow t+1;$

End

$t' \leftarrow \min_t E_n^t; S_n \leftarrow S_n^{t'}; E_n \leftarrow E_n^{t'};$

If ($E_n < E_{\text{best}}$)

$S_{\text{best}} \leftarrow S_n; E_{\text{best}} \leftarrow E_n;$

End

$iter \leftarrow iter+1;$

End

3.2.2. Çözümlerin gösterimi (Solution representation)

Önerilen algoritmada çözümler string olarak gösterilmektedir. 5 işin olduğu bir örnek aşağıda Şekil 2' de verilmiştir. İlk satır makine hızını, ikinci satır işleri göstermektedir. Buna göre, 2. iş ilk sırada çizelgelenmiştir ve 1 makine hızıyla işlem görmüştür. Benzer şekilde, 3 işi 2. sırada çizelgelenmiştir ve 4 makine hızıyla işlem görmüştür.

Makine hızı	1	4	2	1	1
İşler	2	3	1	4	5

Şekil 2. Örnek gösterim (An example representation)

3.2.3. Komşu çözümlerin oluşturulması (Neighborhood generation)

Önerilen algoritmada, komşu çözümlerin oluşturulmasında 2 yöntem ardışık olarak kullanılmaktadır.

Yöntem 1: Yöntem 1 ile rassal olarak bir iş sırası oluşturulur. Tüm işler için makine hızı en yüksek hız kabul edilir. Bu durumda toplam geciken iş sayısı U_k^{best} değerinden daha fazlaysa, yeni bir iş sırası rassal olarak tekrar türetilir. Çünkü işler en yüksek makine hızında işlenmesi durumunda bile U_k^{best} değerinden büyük sayıda iş gecikiyorsa, ilgili sırada U_k^{best} değerinin sağlanamayacağı anlaşılmaktadır. Yöntem 1 en fazla U_k^{best} sayıda iş gecikecek şekilde çözüm türetene kadar rassal olarak iş sırası türetmektedir. En fazla U_k^{best} sayıda iş gecikecek şekilde çözüm türetildikten sonra, bu çözüme Yöntem 2 uygulanmaktadır.

Yöntem 2: Yöntem 1 ile elde edilen çözüm Yöntem 2' nin uygulanmasıyla toplam enerji tüketimi açısından iyileştirilmektedir. Rassal seçilmiş işlerde makine hızı düşürülerek toplam enerji tüketimi iyileştirilmekte fakat U_k^{best} değerini sağlamaya devam eden çözümler elde edilmektedir. Bu amaçla aşağıda verilen özellikler kullanılmaktadır.

İş yükü L_j olan j işi v_q hızıyla işlendiğinde, işin işlem süresi (L_j/v_q) olur. Enerji tüketim miktarı ise ($L_j/v_q \times y_{jq} \times p_{jq}$) olarak hesaplanır. Eğer işlem hızı v_q değerinden $v_{(q-1)}$ değerine düşürülürse, işlem zamanı ($L_j/v_{(q-1)}$) olur ve işlem zamanı artar. Bu durumda enerji tüketim oranı ise ($L_j/v_{(q-1)} \times y_{jq} \times p_{j(q-1)}$) olarak hesaplanır. $p_{j(q-1)} < p_{jq}$ olduğu için enerji tüketimi makine hızı düşürüldüğünde düşebilir. Fakat j işinin işlem süresi arttığı için toplam enerji tüketimi artabilir. En son sırada çizelgelenen ve ardışık geciken işlerin işlem süreleri arttırıldığında toplam geciken iş sayısı değişmez. Fakat toplam enerji tüketimi makine daha yavaş çalıştırılarak azaltılabilir. En son sırada çizelgelenen ve ardışık olarak geciken her bir iş için toplam enerji tüketiminin en küçük olduğu hız, makine hızı olarak seçilir. Eğer bir işin tamamlanma zamanı işin teslim tarihinden küçükse, iş gecikmemektedir. Bir çizelgede her bir iş için $B_j = (d_j - C_j)$

değeri hesaplanır. Ardışık olarak gecikmeyen işlerde en küçük B_j değeri kadar işlerin işlem süreleri arttırılacak şekilde rassal seçilmiş işlerin makine hızları düşürülür. Böylece U_k^{best} değerini sağlayan ve toplam enerji tüketimini en küçükleyen çözümlerin bulunması amaçlanır.

Yöntem 2'nin kullanımında rassallık olduğu için farklı çözümler elde edilebilir. Yöntem 1 sonucu elde edilen çözüme M_{2max} sayıda yöntem 2 uygulanmakta, bu sayıya ulaşıldığında yöntem 2' nin uygulanması durdurulmaktadır. Yöntem 2' nin uygulanması sonucu U_k^{best} değerinden daha fazla sayıda işin geciktiği çizelge sayıları $maxtry$ sayısına ulaştığında da Yöntem 2' nin uygulaması durdurulmaktadır. Yöntem 2' nin uygulaması sonucu elde edilen çözümlerden toplam enerji tüketiminin en küçük olduğu çözüm alınmaktadır. Belirli bir sayıda deneme sonucu U_{best} değerine ulaşılmıyorsa algoritma sonlanır. Komşu çözüm türetme mekanizması aşağıda verilmiştir.

Procedure komşu çözüm türetme yöntemi

Input: Problem parametreleri,

$M_{2max}, maxtry, U_k^{best}$

Output: Komşu çözüm

Rassal bir çözüm oluştur ve bu çözüme

makine hızını en yüksek hız kabul et ve

çözümde U_k geciken iş sayısını bul;

While ($U_k > U_k^{best}$)

Rassal çözüm oluştur ve çözümde U_k

geciken iş sayısını bul;

End

$m \leftarrow 1; try \leftarrow 1;$

While ($m < M_{2max}$ and $try < maxtry$)

Toplam E' yi en küçükleyecek şekilde

işlerin makine hızını revize et ve U_k

değerini hesapla;

If ($U_k > U_k^{best}$)

$try \leftarrow try + 1;$

End

$m \leftarrow m + 1;$

End

4. NSGA II ALGORİTMASI (NSGA II ALGORITHM)

Deb, Pratap, Agarwal ve Meyarivan [38] tarafından 2002 yılında ortaya atılan çok amaçlı bir evrimsel algoritmadır. NSGA II pareto en iyi kümeyi araştırmak için GA tabanlı olarak tasarlanmıştır. NSGA II algoritmasında öncelikle N büyüklüğündeki başlangıç popülasyonu oluşturulur. Çalışma kapsamında başlangıç popülasyonu rassal olarak oluşturulmuştur. Bu popülasyon hızlı baskın olmayan sıralama yaklaşımıyla derecelendirilir. Bu yaklaşımda ise her bireyin baskın olduğu bireyler ve bu bireye baskın gelen bireyler kaydedilir. Eğer bir bireye baskın olan hiçbir birey yok ise bu bireyin n_j değeri sıfıra eşittir ve bu birey birinci baskınlık derecesine aktarılır. Daha sonra birinci baskınlık derecedeki bireylerin bastırıldığı bireylerin n_j değerleri bir azaltılır ve n_j değeri sıfır olan bireyler ikinci dereceye aktarılır. Sonra ikinci derecedeki bireylerin bastırıldığı bireylerin n_j değerleri bir azaltılır ve n_j değeri sıfır olan bireyler üçüncü dereceye aktarılır. Bu işlem tüm bireyler bir dereceye

aktarılan kadar devam eder. Sonra yığındaki bireylere yeniden üretim, çaprazlama ve mutasyon operatörleri uygulanarak Q_t popülasyonu oluşturulur. Yeniden üretim mekanizmasında baskınlık derecelerinden yararlanıldığı için başlangıç popülasyonunun öncelikle baskınlık derecelendirmesi yapılmaktadır. Başlangıç popülasyonu ve Q_t popülasyonunun olduğu $2N$ büyüklüğündeki popülasyon R_t popülasyonu olarak isimlendirilir ve bu popülasyona baskınlık derecelendirmesi uygulanır. Diğer jenerasyona aktarılan bireylerin seçiminde baskınlık derecelendirmesi dikkate alınır [39]. Diğer popülasyonun da birey sayısı N olması istendiğinden $2N$ büyüklüğündeki popülasyondaki tüm bireyler diğer jenerasyona aktarılamamaktadır. Şöyleki baskınlık derecesi en küçük bireylerden başlayarak bireyler $(t+1)$. popülasyona aktarılır. Aktarılan bireylerin sayısının N olduğu son baskınlık derecesine ise yığılma uzaklığı hesabı uygulanır. Çünkü diğer jenerasyonun birey sayısının N olması istendiği için büyük olasılıkla bu baskınlık derecesine sahip tüm bireyler diğer popülasyona aktarılamayacaktır. Buradaki bireyler yığılma uzaklığı dikkate alınarak diğer popülasyona seçilir. Yığılma uzaklığı büyük olan bireyler diğer popülasyona seçilmektedir. Sonuç olarak bir bireyin baskınlık derecesi ne kadar küçükse bireyin diğer popülasyona aktarılma olasılığı o kadar yüksektir. Eşit baskınlık derecesinde olan bireylerde ise bir tercih yapılması gerekiyorsa yığılma uzaklığı büyük olan bireyin yeni jenerasyona aktarılma olasılığı yüksek olmaktadır [40]. Daha sonra yeni jenerasyondan yeniden üretim, çaprazlama ve mutasyon operatörleri ile $Q_{(t+1)}$ popülasyonunu oluşturur ve işlemler nesil sayısı sağlanana kadar devam eder.

4.1. Yığılma Uzaklığı Hesaplaması (Calculation of the Crowding Distance)

Öncelikle her bir amaç için yığındaki bireyler küçükten büyüğe sıralanır ve en küçük ve en büyük değerlere sahip bireylerin yığılma uzaklığı sonsuz alınır. Böylece uç değerlerdeki bireylerin yeni jenerasyona aktarılma olasılığı artırılır. Arada kalan diğer çözümlere, iki komşu çözümün amaç fonksiyonu değerlerinin mutlak normalize edilmiş farklarından bulunan bir uzaklık değeri atanır. Toplam yığılma uzaklığı değeri her amaç için hesaplanan bireysel uzaklıkların toplamıdır. Yığılma uzaklığı hesaplaması aşağıda verilmiştir [40].

Procedure Yığılma uzaklığı sıralaması

Input: F_k (Yığılma uzaklığı hesaplanacak baskınlık derecesindeki bireyler)

Output: I_i (bireylerin yığılma uzaklığı)

$L = |F_k|;$

//her bir amaç fonksiyonuna göre bireyleri küçükten büyüğe sırala

For $j=1:m$

$I_{(j)} = \text{sort}(I_{(j)});$

$I_{1(j)} = I_{L(j)} = 1000000000;$

End

For $i=2:L-1$

For $j=1:m$

$$I_{(j)} = I_{(j)} + |f_{i+1}^j - f_{i-1}^j| / (f_{max}^j - f_{min}^j);$$

$$I_i = I_i + I_{(j)};$$

End

End

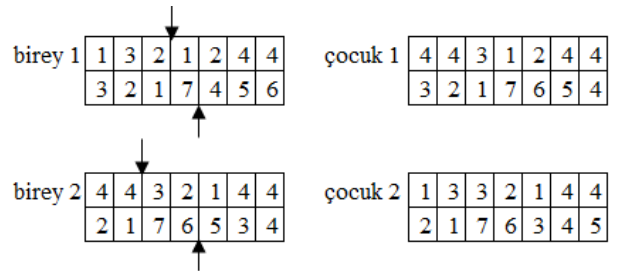
4.2. NSGA II Algoritmasının parametreleri (Parameters of NSGA II)

NSGA II algoritmasının parametreleri yığın büyüklüğü (N), çaprazlama oranı, mutasyon oranı ve jenerasyon sayısıdır. Literatür dikkate alınarak ve yapılan denemeler sonucu yığın büyüklüğü 150, çaprazlama oranı 0,9, mutasyon oranı 0,1 belirlenmiştir. Parametre değerleri Tablo 3' de verilmiştir. NSGA II algoritması önceden belirlenen jenerasyon sayısına ulaştığında durmaktadır. Yani jenerasyon sayısı algoritmanın durdurma kriteridir. Algoritma önerilen sezgisel algoritmayla eşit sürede çalışması amaçlandığından jenerasyon sayısı belirlenmemiştir. Yani algoritma önerilen algoritmaya eşit sürede çalıştığında sonlanmaktadır.

Tablo 3. NSGA II Algoritmasının parametre değerleri
(Parameters of the NSGA II)

Parametre adı	Değeri
Yığın büyüklüğü (N),	150
Çaprazlama oranı	0,9
Mutasyon oranı	0,1

Başlangıç çözüm rassal olarak oluşturulmuştur. Çaprazlamada ise literatürde sıklıkla kullanılan pozisyon tabanlı çaprazlama (position based crossover) yöntemi kullanılmıştır. Buna göre aşağıdaki Şekil 3' de verilen Birey 1 ve Birey 2 çaprazlanacağı zaman Birey 1' in birinci satırı için rassal bir sayı ve ikinci satırı için rassal bir sayı türetilmektedir. Örneğin birey 1 için birinci satır için 3, ikinci satır için 4 sayısı türetilmiştir. Çocuk 1' e, birey 1' in birinci satırı için türetilen genlerden sonraki genler direkt aktarılmıştır. Önceki genler ise Birey 2' den alınmıştır. Çocuk 1' nin ikinci satırı ise birey 1 in ikinci satırı için rassal türetilen gene kadar olan genler direkt aktarılmış, kalan genler ise birey 2' deki iş sırası dikkate alınarak eklenmiştir. Aynı işlem çocuk 2' ye uygulanmıştır. Sonuçta birey 1 ve 2' nin çaprazlaması sonucu şekilde verilen Çocuk 1 ve 2 oluşmaktadır. Çaprazlama sonucu oluşan bireylerde ardışık geciken son işlerin makine hızı en düşük toplam enerji tüketimi sağlanacak şekilde değiştirilmektedir. Böylece toplam geciken iş sayısı değişmemekte ve enerji tüketim oranı azaltılmaktadır.



Şekil 3. Çaprazlama örneği (Example crossover)

Mutasyonda ise rassal olarak seçilmiş bir iş için makine hızı rassal olarak değiştirilmektedir. Yeniden üretim operatöründe ise ikili turnuva yöntemi kullanılmakta olup, rassal seçilmiş iki bireyden derecesi daha küçük olan çaprazlamaya sokulmaktadır.

5. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Tüm algoritmalar MATLAB R2012b' da kodlanmış ve Intel(R) Core (TM) i7-5500 U CPU at 2.40 GH z, 12GB bellekli ve Windows 10 işletim sistemine sahip bir bilgisayarda çalıştırılmıştır. Önerilen matematiksel model GAMS Cplex çözücüsü ile çözülmüştür

5.1. Test Problemlerinin Türetilmesi ve Önerilen Sezgisel Algoritmanın Parametreleri

(Test Problem Generation and Parameters of the Heuristic Algorithm)

Konuyla ilgili test problemlerine ulaşamadığından, önerilen sezgisel algoritmanın başarısının gösterilmesinde literatür dikkate alınarak test problemleri türetilmiş ve bu problemler dikkate alınmıştır. İş sayısı 10, 30, 50, 70 ve 100 olarak alınmıştır. İş yükü L_j ise $U(5,50)$ aralığında düzgün dağılıma uygun türetilmiştir. Hazırlık süreleri s_{ij} , $U(3,15)$ aralığında düzgün dağılıma uygun türetilmiştir. Literatür dikkate alınarak makine hızları 0.75, 1, 1.25 ve 1.50 olarak alınmıştır. İşlerin teslim zamanları ise Lee vd. [39], tarafından önerildiği şekilde türetilmiştir. Teslim tarihi sıklık faktörü τ ve teslim zamanı aralık faktörü R teslim zamanlarının türetilmesinde kullanılmaktadır. τ ve R Eş. 15'de verildiği şekilde hesaplanmaktadır.

$$\tau = 1 - \bar{d} / C_{\max} \quad (14)$$

$$R = (d_{\max} - d_{\min}) / C_{\max} \quad (15)$$

(C_{\max} : Son işin tamamlanma zamanı, \bar{d} : Ortalama teslim tarihi, $d_{\max(\min)}$: maksimum (minimum) teslim tarihi)

Çalışmada τ 0.3 veya 0.7 ve R 0.2 veya 0.8 olarak alınmıştır. Teslim tarihleri τ olasılıkla $[\bar{d} - R\bar{d}, \bar{d}]$ ve $(1 - \tau)$ olasılıkla $[\bar{d}, \bar{d} + (C_{\max} - \bar{d})R]$ aralığında türetilir. C_{\max} için alt sınır Eş. 16'daki gibi bulunmuştur.

$$S_j = \min_{h \in J} s_{hj} \quad \forall j \quad (16)$$

Eş. 16 ile her bir iş için hazırlık süresi en küçük hazırlık süresi olarak alınmaktadır.

$$S^1 = \sum_j S_j \quad (17)$$

S^1 değeri tüm S_j değerlerin toplamı olarak alınmakta olup Eş. 17 ile gösterilmiştir

$$C_{\max} = (\sum_j L_j / v_{q^*} + S^1) \quad (18)$$

Alt sınır ise Eş. 18 ile hesaplanmaktadır. (v_{q^*} : En yüksek makina hızı) [41].

Enerji tüketim oranları Wu ve Che [36], tarafından önerildiği şekilde türetilmiştir. Enerji tüketim oranlarını türetmek için öncelikle p_j parametresi $U[4,18]$ aralığında düzgün dağılıma uygun türetilir. Enerji tüketim oranları ise $p_{jq} = p_j * L_j * v_q^{\alpha-1}$ olarak hesaplanır. $p_j * L_j$ değerleri $v_q^{\alpha-1}$ ile çarpılmakta, böylece yüksek hızda enerji tüketim oranı yüksek olmaktadır. Literatür dikkate alınarak, $\alpha=3$ olarak alınmıştır. Her bir özelliğindeki problemden iki farklı test problem türetilmiş olup, toplam 40 adet test problemi çözülmüştür. Önerilen sezgisel algoritmanın parametreleri belirlenirken yapılan testler dikkate alınmıştır. Bu amaçla aşağıdaki tabloda verilen parametre seviyeleri için denemeler yapılmış ve * ile belirtilen seviyeler parametre değeri olarak alınmıştır.

Tablo 4. Önerilen sezgisel algoritmanın parametreleri
(Parameters of the heuristic algorithm)

Parametre	Değer
maxtry	20*, 30 ve 40
M _{2max}	25, 50* ve 75
IT	100, 200* ve 300
T ₀	500, 1000* ve 1500
q	0.99*, 0.95 ve 0.90
T _{final}	1
maxiter	100, 150 ve 200*

5.2. Performans Ölçütleri (Performance Criteria)

Performans ölçütleri olarak literatürde en sık kullanılan 4 kriter dikkate alınmıştır. Bu kriterler: Bastırılmamış çözüm sayısı (N_{nd}), bastırılmamış çözüm oranı (R_{nd}), referans kümesinden ortalama uzaklık (AD) ve çözüm süresidir [36]. PF_a a yönteminin kullanımı sonucu elde edilen pareto etkin yüzeyi göstermektedir. a algoritmasının başarısı PF_a ' yı RS ile karşılaştırılması sonucu belirlenir. RS (reference set) ise probleme ait optimal ya da optimale yakın pareto yüzeyi göstermektedir. Fakat RS problemin NP- zor yapısından dolayı pek çok test problemi için optimal şekilde bulunamamaktadır. Bu yüzden RS önerilen algoritmanın kullanımı sonucu elde edilen pareto etkin yüzey ile (PF_{her}), NSGA II algoritmasının çalıştırılması sonucu elde edilen pareto etkin yüzeyin (PF_{NSGAI}) birleşimi olarak alınmaktadır. ($RS = PF_{her} \cup PF_{NSGAI}$) Daha sonra RS kümesinde bastırılmış çözümler ayıklanmaktadır ve RS kümesinde sadece bastırılmamış çözümler bırakılmaktadır. $N_{nd}(a)$ ise RS kümesinde olup a yöntemi sonucu bulunan çözümlerin sayısını göstermektedir. $N_{nd}(a)$ değeri ne kadar büyükse algoritma o kadar başarılıdır [42]. $R_{nd}(a) = \frac{N_{nd}(a)}{|PF_a|}$

olarak hesaplanmaktadır. $|PF_a|$, PF_a kümesinin eleman sayısını gösterir. $R_{nd}(a)$ değerinin sıfır olması algoritmanın bulunduğu çözümlerin diğer algoritma tarafından bastırıldığını göstermektedir. Değerin 1 olması ise algoritmanın çok başarılı olduğunu, yani algoritmanın bulunduğu sonuçların diğer algoritmanın bulunduğu sonuçlar tarafından bastırılmadığını gösterir. Bir başka kriter ise AD(a) olup, referans kümesinden ortalama uzaklığı göstermektedir. Bu kriter RS deki bir çözüme en yakın çözümün ortalama normalize uzaklığını göstermektedir. Aşağıda Eş. 19' de gösterildiği gibi hesaplanmaktadır:

$$\frac{1}{|RS|} \sum_{y \in RS} \left\{ \min_{x \in PF_a} \sqrt{\left(\frac{U(x)-U(y)}{\bar{U}-\underline{U}} \right)^2 + \left(\frac{E(x)-E(y)}{\bar{E}-\underline{E}} \right)^2} \right\} \quad (19)$$

\bar{U} ve \underline{U} (veya \bar{E} ve \underline{E}) RS' deki maksimum ve minimum geciken iş sayısı (veya toplam enerji tüketimi) değerlerini göstermektedir. Görüldüğü gibi AD(a), PF(a)'nın RS' ye uzaklığını gösterir ve düşük AD(a) değeri algoritmanın yüksek performansı olduğunu gösterir.

5.3. Karşılaştırma Sonuçları (Comparison Results)

Tablo 5'de ε - kısıt yöntemi, önerilen sezgisel algoritma ve NSGA II algoritmasının literatürdeki performans ölçütlerine

göre karşılaştırılması verilmiştir. 30 ve daha fazla iş olduğu test problemlerinde belli bir ε değerinde 18000 sn de uygun çözüm elde edilemediği için ε - kısıt yöntemiyle sadece 10 iş olduğu test problemleri karşılaştırılmıştır. ε - kısıt yöntemiyle 10 iş olduğu test problemlerin tüm Pareto etkin çözümler bulunmuştur. Önerilen sezgisel algoritma da 10 iş olduğu test problemlerinde tüm Pareto etkin çözümleri bulmaktadır. 10 iş olduğu test problemlerinde önerilen sezgisel algoritma ile R_{nd} değeri 1 bulunmuştur. Pareto etkin çözüm sayısı ise ε - kısıt yöntemiyle aynıdır. 30 ve daha fazla iş olduğu test problemlerinde ise önerilen sezgisel algoritma ve NSGA II algoritması karşılaştırılmıştır. NSGA II algoritmasının bulunduğu pek çok çözüm önerilen sezgisel

Tablo 5. Karşılaştırma sonuçları (Comparison results)

n	τ	R	ε - kısıt yönt.				Önerilen sezgisel				NSGA II			
			N_{nd}	R_{nd}	AD	CPU	N_{nd}	R_{nd}	AD	CPU	N_{nd}	R_{nd}	AD	
10	0.3	0.2	3	1	0	17452	3	1	0	352,21	1	0,25	5,02	
			4	1	0	56852	4	1	0	422,52	2	0,5	6,12	
		0.8	4	1	0	36985	4	1	0	411,2	2	1	11,01	
			4	1	0	46521	4	1	0	366,45	3	0,42	5,13	
	0.7	0.2	3	1	0	42638	3	1	0	223,14	1	0,2	2,21	
			3	1	0	39852	3	1	0	306,41	3	1	1,18	
		0.8	4	1	0	54852	4	1	0	312,14	2	0,4	10,2	
			3	1	0	38966	3	1	0	302,45	2	0,33	9,45	
			0.3	0.2	-	-	-	9	1	0	651,41	2	0,33	12,52
				-	-	-	-	11	1	0	758,42	1	0,16	10,56
30	0.8	-	-	-	-	8	1	0	542,63	3	0,6	14,51		
		-	-	-	-	7	1	0	699,52	0	0	11,89		
		0.2	-	-	-	6	1	0	534,41	1	0,1	12,32		
		-	-	-	-	5	1	0	608,85	0	0	15,41		
	0.7	0.8	-	-	-	5	1	0	501,35	0	0	6,52		
		-	-	-	-	5	1	0	852,74	2	0,25	14,12		
50	0.3	0.2	-	-	-	14	1	0	854,63	3	0,33	13,62		
		-	-	-	-	12	1	0	786,96	4	0,4	13,85		
		0.8	-	-	-	12	1	0	985,41	0	0	21,32		
		-	-	-	-	8	1	0	982,41	0	0	25,41		
	0.7	0.2	-	-	-	8	1	0	993,42	0	0	18,63		
		-	-	-	-	8	0,8	0,05	953,45	2	0,33	10,89		
		0.8	-	-	-	7	1	0	874,65	0	0	11,68		
		-	-	-	-	5	1	0	904,12	1	0,16	15,41		
		0.3	0.2	-	-	-	17	1	0	1206,52	0	0	22,21	
			-	-	-	-	17	1	0	1198,85	3	0,33	20,49	
70	0.8	-	-	-	-	21	1	0	1058,41	0	0	20,2		
		-	-	-	-	20	0,9	0,02	1002,41	4	0,4	25,19		
		0.2	-	-	-	12	1	0	1287,65	0	0	9,88		
		-	-	-	-	10	1	0	1185,13	3	0,2	8,89		
	0.7	0.8	-	-	-	11	1	0	1174,41	0	0	18,84		
		-	-	-	-	15	1	0	1102,1	1	0,08	10,42		
100	0.3	0.2	-	-	-	20	1	0	1478,52	0	0	22,52		
		-	-	-	-	24	0,92	0,5	1585,41	4	0,25	20,19		
		0.8	-	-	-	21	1	0	1563,42	1	0,06	25,57		
		-	-	-	-	20	1	0	1685,13	4	0,19	22,17		
	0.7	0.2	-	-	-	14	1	0	1341,65	4	0,4	12,85		
		-	-	-	-	19	1	0	1585,36	3	0,27	11,1		
		0.8	-	-	-	15	1	0	1574,65	2	0,14	11,41		
		-	-	-	-	16	1	0	1515,78	0	0	10,17		
		Ortalama						11	0,99	0,01	918,158	2	0,23	13,78

algoritmanın bulunduğu çözümlerce bastırılmıştır. Sonuçta önerilen algoritma ile 10 işin olduğu test problemlerinde tüm Pareto etkin çözümler bulunabilmiştir. 30 ve daha fazla sayıda işin olduğu test problemlerinde ise çok amaçlı problemlerin çözümünde sıklıkla kullanılan NSGA II algoritmasına göre performans kriterlerine göre daha başarılı olduğu sonucuna varılmıştır.

6. SONUÇLAR (CONCLUSIONS)

Bu çalışmada, enerji etkin, sıra bağımlı hazırlık süreli tek makine çizelgeleme problemi incelenmiştir. Ele alınan problemde hız ayarlama metodu dikkate alınmıştır. Amaç fonksiyonları toplam enerji tüketiminin ve toplam geciken iş sayısının enküçüklenmesi olup, problem çok amaçlıdır. Probleme özgü çok amaçlı matematiksel model önerilmiştir. Tüm Pareto etkin yüzeyin bulunması için ϵ - kısıt yöntemi kullanılmıştır. Fakat problemin NP- zor yapısı sonucu, pek çok test problemi için ϵ - kısıt yöntemi en iyi çözümü garanti edememektedir. Bu yüzden çok amaçlı sezgisel bir algoritma önerilmiştir. Önerilen algoritmanın başarısı küçük boyutlu problemlerde ϵ - kısıt yönteminin sonucuyla, diğer problemlerde ise NSGA II algoritmasının sonuçlarıyla karşılaştırılarak gösterilmiştir. Sonuçta önerilen çok amaçlı sezgisel algoritma NSGA II algoritmasından daha iyi sonuçlar elde etmiştir. Bu çalışmayla birlikte enerji etkin çizelgeleme problemlerinde ilk defa toplam geciken iş sayısının enküçüklenmesi amacı dikkate alınmıştır.

KAYNAKLAR (REFERENCES)

- Ding J. Y., Song S., Wu C., Carbon- efficient scheduling of flow shops by multi- objective optimization, *European Journal of Operational Research*, 248, 758-771, 2016.
- Che A., Wu X., Peng J., Yan P., Energy- efficient bi-objective single machine scheduling with power-down mechanism, *Computers and Operations Research*, 85, 172- 183, 2017.
- Dai, M., Tang, D., Giret, A., Salido, M., Li, W., Energy-efficient scheduling for a flexible flow shop using an improved genetic- simulated annealing algorithm, *Robotic and Computer Integrated Manufacturing*, 29, 418- 429, 2013.
- Mouzon, G., Yildirim, M. B., A framework to minimise total energy consumption and total tardiness on a single machine, *International journal of Sustainable Engineering*, 105- 116, 2008.
- Mouzon, G., Yildirim, M. B., Twomey, J., Operational methods for minimization of energy consumption of manufacturing equipment, *International Journal of Production Research*, 45 (18-19), 4247- 4271, 2010.
- Yildirim, M. B., Mouzon, G., Single- machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm, *IEEE Transactions on Engineering Management*, 59 (4), 585- 597, 2012
- Gong, X., Pessemier, T. D., Joseph, W., Martens, L., A generic method for energy- efficient and energy- cost-effective production at the unit process level, *Journal of Cleaner Production*, 113, 508- 522, 2016
- Che, A., Wu, X., Peng, J., Yan, P., Energy- efficient bi-objective single- machine scheduling with power- down mechanism, *Computers and Operations Research*, 85, 172- 183, 2017.
- Tarakçı, E., Zaim A.H., Öztaş, O., Design of a genetic algorithmic approach for reducing energy consumption in single machine production systems, *Journal of International Trade, Logistics and Law*, 4 (1), 101, 2018.
- Shrouf, F., Ordieres- Mere, J., Garcia-Sanchez, A., Ortega-Mier, M., Optimizing the production scheduling of a single machine to minimize total energy consumption costs, *Journal of Cleaner Production*, 67, 197- 207, 2014
- Tsao, Y., Thanh, V., Hwang, F., Energy- efficient single- machne scheduling problem with controllable job processing times under differential electricity pricing, *Resources, Conservation and Recycling*, 161, 104902, 2020.
- Zhou, S., Jin, M., Du, N., Energy- efficient scheduling of a single batch processing machine with dynamic job arrival times, *Energy*, 209, 118420, 2020.
- Mikhaylidi, Y., Naseraldin, H., Yedidsion, L., Operations scheduling under electricity time- varying prices, *International Journal of Production Research*, 53 (23), 7136- 7157, 2015
- Aghelinejad, M., Ouazene, Y., Yalaoui, A., Machine and production scheduling under electricity time varying prices, *IEEM*, 2016 IEEE International conference on IEEE, 992- 996, 2016.
- Gong, X., Pessemier, T. D., Joseph, W., Martens, L., An energy- cost aware scheduling methodology for sustainable manufacturing, *Procedia CIRP*, 29, 185- 190, 2015.
- Gong, X., De Pessemier, T., Joseph, W., Martens, L., A power data driven energy-cost-aware production scheduling method for sustainable manufacturing at the unit process level, 2016 IEEE 21 st international conference on IEEE, 1-8, 2016.
- Aghelinejad, M., Ouazene, Y., Yalaoui, A., Preemptive scheduling of a single machine with finite states to minimize energy costs, *Springer International Publishing*, 591-599, 2017.
- Gong, X., Wee, M., Pessemier, T., Verbrugge, S., Colle, D., Martens, L., Joseph, W., Integrating labor awareness to energy- efficient production scheduling under real time electricity pricing: An empirical study, *Journal of Cleaner Production*, 168, 239- 253, 2017
- Lee, S., Chung, B. D., Jeon, H. W., Chang, J., A dynamic control approach for energy- efficient production scheduling on a single machine under time-varying electricity pricing, *Journal of Cleaner Production*, 165, 552- 563, 2017.
- Aghelinejad, M., Ouazene, Y., Yalaoui, A., Production scheduling optimisation with machine state and time-dependent energy costs, *International Journal of Production Research*, 56 (16), 5558- 5575, 2018.

21. Zhang, H., Fang, Y., Pan, R., Ge., C., A new greedy insertion heuristic algorithm with a multi- stage filtering mechanism for energy- efficient single machine scheduling problems, *Algorithms*, 11 (2): 18, 2018.
22. Rubaiee, S., Cinar, S., Yildirim, M. B., An energy-aware multiobjective optimization framework to minimize total tardiness and energy cost on a single machine nonpreemptive scheduling, *IEEE Transactions on Engineering Manegement*, 99, 1- 16, 2018.
23. Artigues, C., Lopez, P., Hait, A., The energy scheduling problem: Industrial case- study and constraint propagation techniques, *International Journal of Production Economics*, 143, 13- 23, 2013.
24. Nattaf, M., Artigues, C., Lopez, P., A hybrid exact method for a scheduling problem with a continuous resoruce and energy constraints, *Constraints*, 20, 304-324, 2015.
25. Liao, X., Zhang, R., Chiong, R., Multi- objective optimization of single machine scheduling with energy consumption constraints, *Computational Intelligence (SSCI)*, 2017 IEEE Symposium series on IEEE, 1-8, 2017.
26. Chen, L., Wang, J., Xu, X., An energy- efficient single machine scheduling problem with machine reliability constraints, *Computers and Industrial Engineering*, 137, 106072, 2019.
27. Albers, S., Fujiwara, H., Energy efficient algorithms for flow time minimization, *ACM Transactions on Algorithms*, 3 (4), 49, 2007.
28. Antoniadis, A., Huang, C. C., Non- preemptive speed scaling, *Journal of scheduling*, 16 (4), 385- 394, 2013.
29. Antoniadis, A., Huang, C. C., Ott, S., A fully polynomial- time approximation scheme for speed scaling with speed state, *Proceedings of the twenty-sixth annual ACM- SIAM symposium on discrete algorithms*, Society for industrial and applied mathematics, 1102- 1113, 2015.
30. Fang, K., Uhan, N. A., Zhao, F., Sutherland, J. W., Scheduling on a single machine under time- of -use electricity tariffs, *Annals of Operations Reserach*, 238, 199- 227, 2016.
31. Tasgetiren, M. F., Öztop, H., Eliiyi, U., Eliiyi D. T., Pan, Q. K., Energy efficient single machine total weigted tardiness problem with sequence- dependent setup times, *International conference on intelligent computing*, Springer, 746- 758, 2018.
32. Seo, D. K., Klein, C. M., Jang, W., Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models, 1. *Computers and Industrial Engineering*, 48 (2), 153- 161, 2005.
33. Chen, W. J., Minimizing number of tardy jobs on a single machine subject to periodic maintenance, *Omega*, 37, 591- 599, 2009.
34. Cyril, B., Samia, O., Minimizing the number of tardy jobs for the single machine scheduling problem: MIP-based lower and upper bounds, *RAIRO*, 47 (1), 33-46, 2013.
35. Liu, Y., Liao, X., An enhanced MOPSO algorithm for energy- efficient single- machine production scheduling, *Sustainability*, 11, 5381, 2020.
36. Aydilek, A., Aydilek, H., Allahverdi, A., Algorithms for minimizing the number of tardy jobs for reducing production cost with uncertain prpcessing times, *Applied mathematical modelling*, 45, 982- 996, 2017.
37. Bootaki, B., Mahdavi, I., Paydar, M. M., A hybrid GA- AUGMECON method to solve a cubic cell formation problem considering different worker skills, *Computers and Industrial Engineering*, 75, 31- 40, 2014.
38. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA II, *IEEE Transactions on evolutionary computation*, 6 (2), 182- 197, 2002.
39. Lee, Y., Bhaskaran, K., Pinedo, M., A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions*, 29 (1), 45-52, 1997.
40. Durmaz, E., Şahin R., NSGA- II and goal programming approach for the multi- objective single row facility layout problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (3), 941- 955, 2017.
41. Bektur, G., Saraç, T., Two parallel injection machine scheduling under crane constraint, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31 (4), 903- 911, 2016.
42. Wu, X., Che, A., A memetic differential evolution algorithm for energy- efficient parallel machine scheduling. *Omega*, 82, 155- 165, 2019.