



Melez Akış Tipi Çizelgeleme Problemi İçin Bir Genetik Algoritma

Abdullah AKTEL¹, M. Mutlu YENİSEY²

Öz

Çizelgeleme, birbiri ile rekabet halindeki bir veya daha fazla hedefi optimize etmek amacı ile sınırlı kaynakların tahsis edilmesine yönelik bir karar verme sürecidir. Klasik çizelgeleme teorisinde problemler öncelikle matematiksel olarak modellendikten sonra optimum sonuç veren algoritmalar ile ya da sezgiseller yardımı ile çözülür. Bu çalışma kapsamında gerek üretim gerekse hizmet sektöründe birçok uygulama alanı olan bir dizi seri operasyonun birden fazla işlem biriminde yapılabildiği melez akış tipi çizelgeleme problemi ele alınmıştır. Problemin çözümüne yönelik olarak bir genetik algoritma tasarlanmış ve algoritmanın performansı çeşitli problem boyutları için analiz edilmiştir. Yapılan simülasyonlar algoritmanın umut verici sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: Melez akış tipi çizelgeleme problemi, genetik algoritma, sezgiseller.

A Genetic Algorithm for Hybrid Flow Shop Scheduling Problem

Abstract

Scheduling is a decision making process that deals with allocation of limited resources with the aim to optimize one or more objectives in competition with each other. In classical scheduling theory problems are firstly modeled mathematically and then solved by the help of exact algorithms or heuristics. In this study, hybrid flow shop scheduling problem which have many application areas in both production and service sectors is discussed. In order to solve the problem, a genetic algorithm was designed and the performance of the algorithm analyzed for the various problems sizes. Empirical results indicate that proposed algorithm offer promising results.

Key Words: Hybrid flow shop scheduling problem, genetic algorithm, heuristics.

¹(TÜBİTAK TÜSSİDE)Türkiye Sanayi Sevk ve İdare Enstitüsü, Kocaeli, abdullah.aktel@tubitak.gov.tr.

². İTÜ, Endüstri Mühendisliği Bölümü, İstanbul, yenisey@itu.edu.tr.

Giriş

Birçok üretim ve hizmet sistemi incelendiğinde yapılan işlerin bir dizi seri operasyondan geçirilerek hazır hale getirildiği görülmektedir. Sıklıkla bu operasyonlar her iş için aynı sırayı izlemek zorundadır. İşlem birimlerinin seri olarak sıralandığı ve işlerinde bu sırayı izlediği işlem ortamı, akış tipi olarak adlandırılır. Bir m- akış tipi işlem ortamında, m- işlem aşaması vardır. İşler mutlaka her aşamada işlem görmeli ve aynı sırayı takip etmelidirler. İşlerin farklı aşamalarındaki işlem süreleri farklı olabilir. Eğer her aşamada sadece bir işlem birimi var ise akış tipi sistem, birden fazla işlem birimi var ise melez (esnek) akış tipi sistem olarak adlandırılır. İşlerin hangi sıra ile ve hangi makinede işleme gireceği ise melez akış tipi çizelgeleme problemi olarak adlandırılır.

Mevcut yazın incelendiğinde melez akış tipi çizelgeleme problemine yönelik olarak çeşitli çözüm algoritmalarının geliştirildiği görülmektedir. Önerilen algoritmalar problemi küçük boyutlu durumlar için optimize etmeye çalışmakta ya da önerilen algoritmaların büyük problem boyutları için optimuma ulaşmayı amaçlamayan nispeten iyi çözümler veren sezgisel algoritmalar oldukları görülmektedir. Önerilen çözüm algoritmaları incelendiğinde genetik algoritmaların bir çok araştırmacı tarafından melez akış tipi çizelgeleme problemine yönelik olarak önerildiği görülmektedir. Önerilen genetik algoritmalar incelendiğinde, çözüm adayları ve kromozom yapısı gereği problemi ya aşama aşama ele alarak çizelgeledikleri ya da ilk aşamada üretilen iş sırasından yola çıkarak diğer aşamaların önce biten işin ilk boşalan işlemcide işlem görmesi prensibine göre çizelgelendikleri görülmektedir.

Bu çalışma kapsamında önerilen genetik algoritma her aşamayı kendi içinde ele almakla beraber çizelgeleme bütün aşamaları göz önünde bulundurularak yapılmıştır. Önerilen genetik algoritmanın performansı analiz edildiğinde ise memnun edici sonuçlar verdiği görülmüştür.

Çalışmanın geri kalan kısmı ise şu şekildedir: ikinci bölümde ele alınan problemle ilgili literatür taraması verilmiştir. Çalışmanın üçüncü kısmında önerilen algoritma detaylı olarak açıklanmış ve algoritmanın verdiği sonuçlar paylaşılmıştır. Son bölümde ise elde edilen bulgular değerlendirilmiş ve ileride yapılacak çalışmalar hakkında öneriler sunulmuştur.

Literatür Taraması (Literature Review)

Esnek akış tipi çizelgeleme probleminin çözümü için uygulanan yaklaşımlar üç ana başlıkta sınıflandırılabilir; analitik yöntemler, sezgiseller, metasezgisel yaklaşımlar.

Analitik Yöntemler (Analytical Methods)

Esnek akış tipi çizelgeleme problemleri için en çok kullanılan analitik yöntem dal sınır algoritmasıdır (Ruiz ve Rodriguez, 2010). Gupta ve diğ. (1997) ise daha düşük alt sınır bularak arama uzayını sınırlandıran bir dal sınır algoritması önermişlerdir. Kim ve diğ. (1997) Johnson kuralı benzeri bir optimal prosedür geliştirerek iki aşamalı melez akış tipi çizelgeleme probleminin üretim süresini minimize etmeye çalışmışlardır. Brockmann ve diğ. (1997) paralel bir dal sınır algoritması öne sürmüşler. Öne sürdükleri algoritma birçok durumda diğer algoritmalara oranla daha hızlı sonuç verme açısından lineer bir iyileşme sağlamıştır. Moursli ve Pochet (2000) esnek akış tipi çizelgeleme probleminin en genel hali için (m-aşama ve her aşama için k-makine durumu) yeni bir dal sınır algoritması ile yeni bir dallanma metodu önermişlerdir. Öncelikle iş bağımlı kuyruk kavramını tanımlamışlardır, buna göre bir işin son iş son aşamadan geçip tamamlanana kadar üretim hattında geçirdiği minimum süre iş bağımlı kuyruk olarak tanımlanmıştır. Geliştirdikleri algoritmada her dallanma düğümü iki ayrı problemde oluşmakta ve iş ayrılış zamanı ve iş bağımlı kuyruk değişebilmektedir. Alt sınırı, ana problemi tek aşamadan oluşan alt problemlere esneterek hesaplamaya çalışmışlardır.

Carlier ve Neron (2000) m-aşama ve her aşama için k-makine durumu için yeni bir dal sınır algoritması önermişlerdir. Hesaplama süresini azaltmak için enerjik düşünme (energetic reasoning) and küresel operasyonlar (global operations) kavramlarını kullanmışlardır. Haouari ve diğ. (2006) önerdikleri dal sınır algoritması iki aşamalı esnek akış tipi çizelgeleme problemini büyük problem boyutları için çözebilmeyi amaçlamışlardır. Allaoi ve Artiba (2006)'nın iki aşamalı esnek akış tipi çizelgeleme problemi için geliştirdiği dal sınır algoritması ile yeni bir dallanma planı önermişleridir. Bu yeni algoritmayı makinelerin üretim için müsait olmadığı durumların önceden bilindiği durumlar için uygulamışlardır. Buna göre her makine en fazla bir periyot müsait olmamaktadır.

Guirchoun ve diğ. (2005) iki aşamalı esnek akış tipi çizelgeleme probleminin bekleme olmaksızın ve ilk aşamada her işin aynı işlem süresinde olduğu durum için bir polinomial zaman algoritması

önermişlerdir. Bu algoritma öncelikle işleri 3 kümeye böler ve daha sonra en kısa işlem süresi kuralına göre sıralar ve her işi ilk müsait makine kuralına göre atar. Sawik (2000) esnek akış tipi çizelgeleme probleminin en genel halinin optimum çözümünü bulmak için bir karma tam sayılı programlama önermiştir. Siparişe göre üretim ve limitli ara stok durumu için esnek akış tipi çizelgeleme probleminin üretim süresini CPLEX çözücü ile minimize etmeye çalışmıştır. İki amaçlı esnek akış tipi çizelgeleme probleminin özdeş makineler ve limitli ara stok durumu için bir leksikografik yaklaşım önermiştir.

Dessouky ve diğ. (1998) üç aşamalı esnek akış tipi çizelgeleme probleminde özdeş makineler durumu ile ilgilenmişler ve üretim süresini minimize ederek optimum sonucu bulan bir dal sınır algoritması önermişlerdir. Sawik (2005;2006), esnek akış tipi çizelgeleme probleminin en genel durumu için makinelerin birden çok işi aynı anda işleyebildiği durum için optimum çözüm veren bir yöntem önermiştir. He ve diğ. (2007) iki aşamalı esnek akış tipi çizelgeleme problemi için ilk aşamada m-özdeş makine ikinci aşamada ise parti işleyici (batch processor) bulunduğu durum için n-iş çizelgeleyen bir dinamik programlama algoritması önermişlerdir.

Sezgisel Yöntemler (Heuristic Methods)

Literatürde esnek akış tipi çizelgeleme problemi için bir çok sezgisel yöntem önerilmiştir, bu sezgiseller arasındaki en basit tip sezgiseller, sevk kuralları (dispatching rules) ya da diğer bir deyişle çizelgeleme politikaları veya liste çizelgeleme algoritmalarıdır. Bu kurallar temel olarak işlerin sıralanması ve makinelere atanmasını içerirler. Guinet ve diğ. (1996) iki aşamalı esnek akış tipi çizelgeleme problemi için üretim süresini minimize eden iki aşamalı bir algoritma önermişlerdir. Problemi öncelikle parçalara ayırmışlar (decompose) ve bir öncelik listesine göre işleri sıralamışlar ve işleri bu öncelik sırasına göre müsait olan makinelere atamışlardır. İşleri derecelendirmek için bir çok sıralama kuralını gözden geçirmişlerdir buna göre Johnson kuralı en iyi sonuç verendir.

Guinet ve diğ. (1996) m-aşamalı bir esnek akış tipi çizelgeleme problemi için maksimum gecikmeyi veya maksimum tamamlanma zamanını minimize etmeye çalışmışlardır. İşleri akış tipi çizelgeleme algoritması kullanarak sıralayıp önceliklendirmişlerdir. Bu algoritmalar içinde Nawaz ve diğ. (1983)'nin sezgiseli (NEH) bütün kriterlere göre diğerlerinden daha üstün sonuç vermektedir. Gupta ve diğ. (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için (ilk aşamada birden çok makine, ikinci aşamada tek makine durumu için) birçok yapıcı sezgiseli incelemiş ve üç aşamalı bir sezgisel önermişlerdir. Buna göre ilk aşamada işler bir Johnson kuralı

uyarlamasına göre sıralanır, ikinci olarak işler sanki ilk aşamada tek makine ikinci aşamada birden çok makine var durumu varsayılarak atanır, son olarak orjinal problemle varsayılan probleme göre yapılan atama sonucunda boş zaman dilimi kalacağından ikinci aşamadaki işlerin tekrardan çizelgelenmesi ilk aşamadaki bitiriş zamanlarına göre yapılır.

Li ve diğ. (1997) iki aşamalı esnek akış tipi çizelgeleme problemini hazırlık sürelerinin olduğu durum için incelemişlerdir. Benzer tipteki işleri, iş aileleri şeklinde gruplamışlar ve siparişler hazırlanırken aile tabanlı hazırlık sürelerini göz önünde bulundurmışlardır. Buna göre ileriye doğru ve geriye doğru tahsis politikalarının performansını sekiz sıralama kuralı ile kombine ederek değerlendirmişler ve geriye doğru tahsis politikasının daha iyi sonuçlar verdiğini gözlemlemişlerdir. Zhang ve diğ. (2005) iki aşamalı esnek akış tipi çizelgeleme problemini hazırlık sürelerinin var olduğu durum için ele almışlardır. Her iş için o işin alt parti büyüklüklerini ve sayısını tespit ederek üretim süresini minimize etmeye çalışmışlardır. Önerdikleri sezgisel öncelikle her iş için parti büyüklüklerini tespit edip birinci aşamadaki toplam iş yükünü her iki aşamaya yaymaya dayanmaktadır.

Oguz ve Ercan (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için geliştirdikleri sezgisel ile problemi aşamalara bölüp her aşamadaki işleri atamak için çizelgeleme kuralları önermişlerdir. 5 çizelgeleme kuralının beraber çalıştığı kombine bir yöntem geliştirmişlerdir. Oguz ve diğ. (2003) iki aşamalı esnek akış tipi çizelgeleme problemi için önerdikleri sezgisel önce işleri literatürdeki bir öncelik kuralına göre sıralamak daha sonra her iş için çizelgeyi bulmak şeklindedir. Vandevelde ve diğ. (2005) önerdikleri ileri ve geri aşama algoritması (up-stage and down-stage algorithm) ile her aşamayı paralel makine çizelgeleme problemi gibi ele alıp çözmüşlerdir. Gupta ve Tunc (1998) iki aşamalı esnek akış tipi çizelgeleme problemi için iki sezgisel önermişlerdir. Buna göre ilk sezgisel işleri erken teslim tarihi (EDD) kuralına göre çizelgeler ve ardından işleri makinelere atar.

Gupta ve diğ. (2002) esnek akış tipi çizelgeleme problemi ve kontrol edilebilir işlem zamanları ve teslim tarihleri durumu için ekleme tabanlı bir algoritma önermişlerdir. Thornton ve Hunsucker (2004) aynı problemi ele almışlardır. Önerdikleri sezgisel bir çok alternatif çizelge üretilip bunların arasından üretim süresini minimum yapacak olanı Johnson kuralını kullanarak seçmektedir. Soewandi ve Elmaghraby (2001) üç aşamalı esnek akış tipi çizelgeleme problemi için dört sezgisel önermiştir, bunlardan üç tanesi işleri bölmeye dayalı bir yaklaşım sergiler. İlk olarak ilk iki aşama ele alınır ve daha sonra üçüncü aşama değerlendirilir. Dördüncü sezgisel ise bu üç aşamalı problem için en doğru iş sırasını bulmak ve makinelere atamak

için kullanılır. Cheng ve diğ. (2001) darboğazların yer değiştirmesini esas alan bir yaklaşım önermişlerdir. Öncelikle k-aşamalı esnek akış tipi çizelgeleme problemi alt problemlere ayrılır ve her aşama kendi içinde tek tek çözülür. Daha sonra ise her aşama şimdiye kadar bulunmuş en iyi çizelge ile kıyaslanarak yeniden çizelgelenir.

Kurz ve Askin (2003) esnek akış tipi çizelgeleme problemini hazırlık zamanları var olduğu durum için ele almışlardır. Üç grup sezgiseli kıyaslamışlardır: işlerin makinelere basit bir şekilde atanması prensibine dayalı periyodik sezgisel, NEH sezgiselinin bir uyarlaması olan ekleme tabanlı sezgisel ve Johnson kuralı tabanlı sezgisel. Sonuç olarak Johnson kuralı sezgiselinin gerçek zamanlı karar verme durumu söz konusu olduğunda iyi bir alternatif olduğuna karar vermişlerdir. Soewandi ve Elmaghraby (2003) iki aşamalı esnek akış tipi çizelgeleme problemi için 3 prosedür önermişlerdir. Birinci sezgisel, simetrik işlem zamanları söz konusu olduğu durumda diğer iki sezgisel ise simetrik olmayan işlem zamanlarının söz konusu olduğu durumlar için iyi sonuç vermektedir. Suresh (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için üç aşamalı bir prosedür önermiştir. İlk olarak aşama birdeki işler çizelgelenir, ikinci olarak işlerin ayrılış zamanları hesaplanır üçüncü olarak ise ikinci aşamadaki işler ayrılış zamanlarına göre çizelgelenir.

Metasezgisel Yöntemler (Metaheuristic Methods)

Basit deterministik sezgisellere rastsallığın eklenmesi ile metasezgisel yaklaşımlar elde edilmiştir. Esnek akış tipi çizelgeleme probleminin çözümünde sıkça kullanılan sezgiseller Tavlama Benzetim, Tabu Arama, Genetik Algoritmalar, Karınca Kolonisi Optimizasyonu, Yapay Sinir Ağları ve Yapay Bağışıklık Sistemi sezgiselleridir. Bu sezgiseller genel olarak arama uzayındaki alternatif permutasyonların daraltılması prensibine dayanır. Temel fikir n adet işi makinelere atayacak en doğru permutasyonu bulmaktır (Ruiz ve Rodriguez, 2010). Haouari ve M'Hallah (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için bir tabu arama ve tavlama benzetim algoritması önerdiler. Önerdikleri tabu arama algoritması çözdükleri problemlerin %35'inde optimum sonuç vermiştir.

Nowicki ve Smutnicki (1998) esnek akış tipi çizelgeleme problemi için mevcut çözümün yakın çevresini ekleme hareketleri (insertion moves) ile keşfeden bir tabu arama algoritması önermişlerdir. Wardono ve Fathi (2004) esnek akış tipi çizelgeleme problemi ve ara stok alanının kısıtlı olduğu durum için bir tabu arama algoritması önermişlerdir. Bu algoritmanın en önemli özelliği arama uzayındaki olası permutasyon

vektörlerini limitlendirebilmesidir. Çizelgenin son halini oluşturmak için yapıcı bir prosedür kullanır. Garcia ve Lozano (2005) iki aşamalı bekleme olmaksızın esnek akış tipi çizelgeleme problemi için bir tabu arama algoritması önermişlerdir. Logendran ve diğ. (2006) hazırlık zamanlarının bulunduğu durum için esnek akış tipi çizelgeleme problemine önerdikleri tabu arama algoritması iki arama düzeyinden oluşmaktadır. İlk olarak en iyi iş ailesi sırasının belirlenmesi daha sonra her bir iş ailesi için en iyi iş sırasının bulunmasıdır.

Akrami ve diğ. (2006) esnek akış tipi üretim ortamında çok ürünli parti büyüklüğü belirleme ve çizelgeleme problemi için bir tabu arama algoritması önermişlerdir. Jin ve diğ. (2006) esnek akış tipi çizelgeleme problemi için iki tavlama benzetim algoritması önermişlerdir. Bu algoritmalar birbirlerinden işleri atamak için kullandıkları kurallarla ayrılırlar. Janiak ve diğ. (2007) esnek akış tipi çizelgeleme problemi için tavlama benzetim, tabu arama ve bu iki metodu beraber kullanan melez bir algoritma önermişlerdir. Melez algoritma diğerlerine göre daha iyi sonuç vermiştir. Tavlama benzetim özellikle esnek akış tipi çizelgeleme probleminde hazırlık sürelerinin olduğu durum için kullanılmıştır (Ruiz ve Rodriguez, 2010). Low (2005) hazırlık sürelerinin bulunduğu durum için toplam akış süresini minimize eden bir tavlama benzetim algoritması önermiştir. Allahverdi ve Al-Anzi (2006) esnek akış tipi çizelgeleme problemi için bir çok sezgiselle beraber tavlama benzetim algoritmasını da önermişlerdir. Bir çok başka çalışmada esnek akış tipi çizelgeleme problemini çözmek için genetik algoritmalar kullanılmıştır.

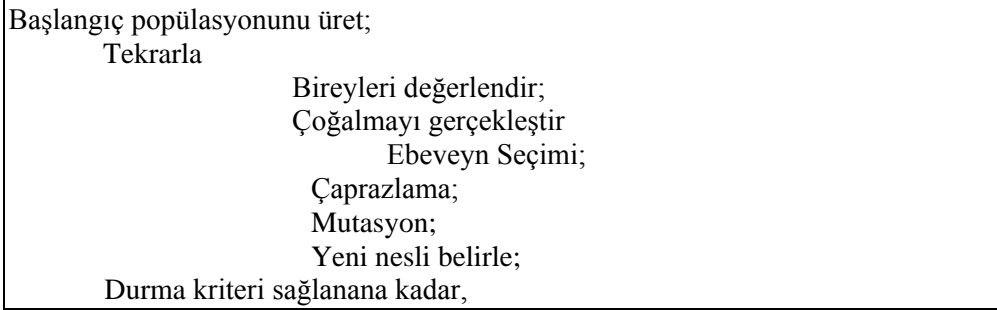
Jin ve diğ. (2002) devre panosu üreten bir montaj hattında üretim süresini bir genetik algoritma kullanarak minimize etmeye çalışmışlardır. Oguz ve Ercan (2005) ara stok alanı kısıtlı durumda esnek akış tipi çizelgeleme problemi için geliştirdikleri genetik algoritmada bu problemi daha iyi açıklayabilmek için yeni bir çaprazlama operatörü kullanmışlardır. Oguz ve diğ. (2003) iki aşamalı esnek akış tipi çizelgeleme problemi için yer değiştirmeli bir paralel genetik algoritma önermiştir. Bu genetik algoritmada popülasyon alt gruplara bölünmüş ve her alt grup bir ardışık genetik algoritma ile çözülmesi amaçlanmıştır. Daha sonra alt popülasyonlar arasında bireylerin yer değiştirmesi ile çeşitlilik sağlanır. Paralel genetik algoritma ile hesaplama süresi düşürülmüş ve çözüm kalitesi artırılmıştır.

Belkadi ve diğ. (2006) de esnek akış tipi çizelgeleme problemi için yer değiştirmeli bir paralel genetik algoritma önermişlerdir. Bu algoritmanın ardışık uygulamalı versiyondan daha iyi sonuç bulunduğunu göstermişlerdir. Ying and Lin (2006) esnek akış tipi çizelgeleme problemi için bir karınca kolonisi sistemi sezgiseli önermişlerdir. Alisantoso ve diğ. (2003) ve Engin

ve Doyen (2004) esnek akış tipi çizelgeleme problemini çözmek için yapay bağışıklık sistemi algoritmasını önermişlerdir. Gupta ve diğ. (2000) iki aşamalı esnek akış tipi çizelgeleme problemini çözmekte kullanılacak en iyi sezgiseli seçmek için yapay sinir ağı kullanmışlardır. Wang ve diğ. (2003) esnek akış tipi çizelgeleme problemi için bir yapay sinir ağı önermişlerdir. Her iterasyonda mevcut alt sınır ile yeni bulunan çözümü kıyaslayarak çalışan bir prosedürdür. Ağırlıkların modifikasyonu alt sınır ile mevcut çözüm arasındaki farkın bir fonksiyonu şeklinde yapılır. Tavakkoli-Moghaddam ve diğ. (2009) aynı problem için iç içe geçmiş değişken komşu aramalı bir memetik algoritma önermişlerdir. Alaykyran ve diğ. (2007) esnek akış tipi çizelgeleme problemi için bir karınca kolonisi optimizasyonu algoritması önermişlerdir. Kurz ve Askin (2004) esnek akış tipi çizelgeleme problemi için bir rastsal anahtar genetik algoritması önermişlerdir. Bu algoritma rastsal sayıları çözümü oluşturacak anahtarları sıralamak için kullanmıştır. Zandieh ve diğ. (2006) aynı problem için önerdikleri yapay bağışıklık sistemi algoritması rastsal anahtar genetik algoritmasından daha iyi çözüm vermektedir.

Önerilen Genetik Algoritma (Proposed Genetic Algorithm)

Önerilen genetik algoritmanın akış diyagramı aşağıdaki gibidir:



Şekil 1. Genetik Algoritma Akış Diyagramı

İlk popülasyonun üretilmesi sonrasında genetik operatörler (çaprazlama, mutasyon) gerçekleştirilir. Yeni nesil belirlenir, ve genetik operatörler uygulanarak durma kriteri sağlanana kadar devam edilir.

Genetik algoritmanın kromozom yapısının permutasyon tabanlı gösterimle temsil edilmesinin uygun olacağı düşünülmüştür. Buna göre her işlem aşaması önünde bekleyen iş sıralarından oluşan kromozom bir çözüm adayını oluşturur. Örneğin iki aşamalı bir sistemde işlem göreceğ beş işin {J1,J2,J3,J4,J5}, ilk aşamadaki işlem sırası {J2,J1,J5,J4,J3}, ikinci aşamadaki işlem sırası ise {J1,J3,J2,J5,J4} şeklindedir. Kromozom ise bu iş

sıralarının birleşimi $\{\{J2, J1, J5, J4, J3\}, \{J1, J3, J2, J5, J4\}\}$ şeklinde ifade edilebilir. Daha sonrasında ise genetik algoritma operatörleri (çaprazlama ve mutasyon) çözüm adaylarına uygulanarak çözümlerin evrilmesi ve daha iyi çözümlere ulaşılması amaçlanır. Bu çalışma kapsamında Cmax (en büyük işlem zamanı) amaç fonksiyon olarak seçilmiştir. Çözüm adaylarına ait uygunluk değerleri hesaplanırken o çözüm permutasyonunda en son tamamlanan işin bitirilme zamanı o çözüm adayının uygunluk değeri olarak alınacaktır. Çözüm adaylarının aldığı uygunluk değerlerinin hesaplanması, ele alınan probleme göre tasarlanan sistemin simüle edilmesi ile elde edilir.

Başlangıç Popülasyonunun Üretilmesi (Initial Population Generation)

Çözüm adaylarının işlerin aşamalara göre permutasyonlarından oluşturulmasına karar verilmiştir. Bunun ise, rassal anahtar temsil (random key representation) ile sağlanabileceği düşünülmüştür. Rassal anahtar temsil yönteminde bir çözüm adayı reel sayılardan oluşan bir vektör olarak düşünülebilir. Vektörün her bir üyesi farklı bir işi simgeler. Örneğin 9 işlik bir çözüm;

$$[0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36] \quad (1)$$

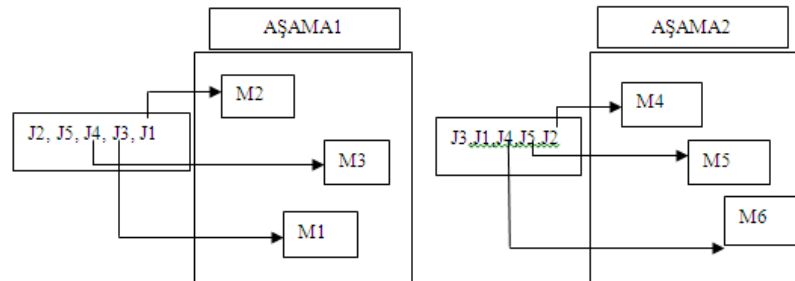
şeklinde olur. Bu vektördeki reel sayıları küçükten büyüğe doğru sıraladığımızda 0.18 1. işi, 0.83 ise 9. işi temsil eder. Buna göre bu 9 işlik çözüm permutasyonu;

$$[2-7-1-5-9-8-4-6-3] \quad (2)$$

şeklinde olur. İlk popülasyonun ise işlerin aşamalara göre rastgele permutasyonlarından oluşturulmasına karar verilmiştir. Bunun sebebi ise rassallığı artırarak yerel optimuma takılmanın önüne geçilmek istenmesidir.

Bireylerin Çözüm Adaylarının Değerlendirilmesi (Simulating Solution Candidates)

Probleme yönelik olarak tasarlanan sistemde çözüm adaylarının (bireylerin), alacakları uygunluk değerleri simüle edilerek bulunmaya çalışılır.



Şekil 2. Melez akış tipi çizelgeleme problemi

Yukarıdaki sistemde görüleceği üzere, 1.aşamının önündeki çözüm adayında işler $\{J2, J5, J4, J3, J1\}$ şeklinde sıralanmışlar. Ele alınan problemde bir iş o aşamadaki her işlem biriminde (M1, M2, M3, M4, M5, M6) işlem görebilmektedir, ancak aşamalardaki işlem birimlerinin işlem hızları birbirlerinden farklıdır. Buna göre permutasyonda ilk sırada bulunan iş (J1), o anda boşta bulunan ve en hızlı işlem kapasitesindeki işlem biriminde işlem göreceği şekilde hareket eder. Daha sonra permutasyonda ikinci sıradaki iş (J3) yine o anda boşta bulunan ve en hızlı işlem birimine atanır. Diğer aşamalar içinde aynı şekilde devam edilerek işler permutasyon sıralarına göre en hızlı boş işlem birimine atanır.

Ebeveyn Seçimi (Parent Selection)

İlk popülasyon üretildikten sonra, popülasyon üyeleri uygunluk değerlerine göre ele alınır ve uygunluk değeri daha iyi çözüm adayları üzerinden genetik operatörler (çaprazlama-mutasyon) gerçekleştirilir. Turnuva seçim yöntemi yardımı ile popülasyondaki daha iyi bireyler ebeveyn olarak seçilir. Turnuva seçim yöntemi m üyeli bir popülasyonun seçilen herhangi bir k üyesinin uygunluklarına göre kıyaslanmasını temel alır. Turnuva Seçim yöntemi, Şekil 3 de verilmiştir.

```

Başla
şu anki üye numarası = 1;
while (şu anki üye numarası ≤ m )
do
    rastgele k birey seç;
    k birey içinden en iyisini seç;
    bu bireyi i. birey olarak belirle;
    çiftleşme havuzu[şu anki üye] = i;
    şu anki üye = şu anki üye + 1;
end do
end while
m: popülasyon büyüklüğü, k:turnuva boyutu

```

Şekil 3. Turnuva seçim yöntemi algoritması

Turnuva seçim yöntemi ile rassallığın artırılması amaçlanmıştır.

Çaprazlama (Cross-over)

Yeni bireyler anne ve babadan gelen genlerin ağırlıklı ortalaması alınarak oluşturulur.

$$\text{Çocuk1} = \alpha x + (1 - \alpha)y \quad (3)$$

$$\text{Çocuk2} = \alpha y + (1 - \alpha)x \quad (4)$$

Bireylere ait genler rasyonel sayılar ile temsil edildiği için bu yönteme başvurulmuştur.

Buna göre 9 işlik baba ve anne bireyleri aşağıdaki gibi ise;

$$\text{Baba} = [0.12, 0.27, 0.33, 0.40, 0.59, 0.64, 0.71, 0.88, 0.95] \quad (5)$$

$$\text{Anne} = [0.34, 0.22, 0.36, 0.28, 0.81, 0.02, 0.16, 0.89, 0.11] \quad (6)$$

$\alpha = 0.5$ için ise

$$\text{Çocuk1} = [0.23, 0.245, 0.345, 0.34, 0.7, 0.33, 0.435, 0.885, 0.53] \quad (7)$$

$$\text{Çocuk2} = [0.23, 0.245, 0.345, 0.34, 0.7, 0.33, 0.435, 0.885, 0.53] \quad (8)$$

Özdeş iki çocuk oluşur. Bu çalışma kapsamında, baba ve annenin çözüme katkısının eşit olmasının uygun olacağına karar verilmiştir.

Mutasyon (Mutation)

Çözüm adayının genlerine rassal bir değer eklenir. Genellikle bu değer ortalaması 0 standart sapması mutasyon olasılığı (pm) olan bir normal dağılımdan üretilir. Buna göre;

$$[0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36] \quad (9)$$

bireyinin genlerine, bu bireyin genlerinin olurlu aralığı (0,1) arasında olacak şekilde rastgele sayılar eklenir. Mutasyon sonrasında oluşan yeni birey;

$$[0.417, 0.878, 0.192, 0.617, 0.939, 0.813, 0.633, 0.479, 0.505] \quad (10)$$

şeklindedir.

Bu çalışma kapsamında gerçekleştirilen mutasyon operatörü uygulamalarında, mutasyon uygulaması sonrasında gen değerlerinin 0 ile 1 arasında değer alacak şekilde olmasını sağlayan, düzgün dağılımdan gelen rastgele sayıların gen değerlerine eklenmesine karar verilmiştir.

Yeni Neslin Belirlenmesi (Survivor Selection)

Ebeveynlerden ve çocuklardan oluşan genişletilmiş birey havuzundan hangilerinin bir sonraki neslin popülasyonunu oluşturacağına karar verilmesi gerekmektedir. Yeni neslin belirlenmesinde uygunluk (fitness) tabanlı stratejiden yararlanılması planlanmıştır. Çözüm adaylarının uygunluk değeri o çözüm permutasyonundaki işlerden tamamlanma zamanı en büyük olan işin bitirme zamanı olarak belirlenmiştir (Cmax). Uygunluk tabanlı stratejide ebeveynlerin ve çocukların birleşimi sonucu oluşan havuzdaki (pool) en iyi bireylerin bir sonraki jenerasyona aktarılması amaçlanır. Buna göre oluşturulan birey havuzundaki tüm bireyler, uygunluk değerlerine göre sıralanır ve en iyi uygunluk değerine sahip olanlar bir sonraki popülasyonun bireyleri olarak belirlenirler. Bu sayede popülasyon ortalamasının daha hızlı bir şekilde iyileşmesi amaçlanır. Özellikle çok geniş ve duplikasyonun olmadığı popülasyonlarda bu stratejinin kullanılması uygun görülmektedir. Bu çalışma kapsamında çizelgelenecek iş sayısına herhangi bir sınır konulması düşünülmemektedir dolayısı ile permutasyonların duplike permutasyon olması olasılığı çok düşüktür. Böyle geniş bir popülasyonun olabildiğince hızlı bir şekilde iyi çözümlere evrilmesini sağlamak için uygunluk tabanlı strateji seçilmiştir.

DENEY TASARIMI VE UYGULAMA SONUÇLARI (EXPERIMENTAL DESIGN AND APPLICATION RESULTS)

Deneysel Tasarım (Experimental Design)

Bu kısım melez akış tipi çizelgeleme probleminin çözülmesine yönelik olarak tasarlanan algoritmanın simülasyonu için yapılan deney tasarımı sürecinin detaylarını içermektedir. Bu çalışma kapsamında sistemdeki işlere ait işlem zamanlarının $P_i \sim U(1,40)$ 1 ile 40 arasında tam sayı değerler alan düzgün dağılımdan üretilmesine karar verilmiştir. İlk popülasyonun birey sayısı yani popülasyon büyüklüğü 100 olarak seçilmiştir. Çaprazlama olasılığı %100'dür, yani bütün anne-baba çözümler çaprazlanacaktır. Mutasyon olasılığının ise; ise;

$$p_m = RN / İŞ_SAYISI \quad (11)$$

0 ile 1 arasındaki bir düzgün rassal sayının, o aşamada bulunan toplam iş sayısına bölünerek bulunmasına karar verilmiştir

Önerilen Genetik Algoritmanın Uygulanması (Application Of Proposed Genetic Algorithm)

Ele alınan probleme yönelik olarak literatürdeki genetik algoritmalar incelendiğinde, çözüm adayları ve kromozom yapısı gereği problemi ya

aşama aşama ele alarak çizelgeledikleri ya da ilk aşamada üretilen iş sırasından yola çıkarak diğer aşamaların önce biten işin ilk boşalan işlemcide işlem görmesi prensibine göre çizelgelendikleri görülmektedir.

Bu çalışma kapsamında önerilen genetik algoritma her aşamayı kendi içinde ele almakla beraber, çözüm adaylarının uygunluk değerleri bütün aşamalar göz önüne alarak hesaplanmakta böylece çizelgeleme yapılırken probleme bir bütün olarak yaklaşılmaktadır.

Önerilen algoritmanın performansını değerlendirebilmek için öncelikle küçük bir problem ele alınmış, ele alınan problemin optimum çözümü ile önerilen algoritmanın verdiği çözüm kıyaslanmıştır. Sonrasında ise algoritma daha büyük problem boyutları için test edilmiştir.

Ele alınan ilk problemde çizelgelenecek iş sayısı 10, işlem aşaması sayısı 2 ve her aşamadaki işlem birimi sayısı 5 olarak seçilmiştir. Aşamalardaki işlem birimleri özdeş değildir, dolayısı ile bir işin aşamalar için farklı işlem birimlerindeki işlem süreleri birbirlerinden farklıdır. Belirlenen parametreler doğrultusunda üretilen ilk problemdeki işlerin aşamalara ve işlem birimlerine göre işlem zamanları:

$$\begin{aligned} \text{İş1} &= \{ \{1,17,19,36,3\}, \{11,5,32,22,32\} \} \\ \text{İş2} &= \{ \{16,31,6,23,10\}, \{14,31,11,21,24\} \} \\ \text{İş3} &= \{ \{29,4,25,16,24\}, \{33,34,17,33,35\} \} \\ \text{İş4} &= \{ \{26,38,23,37,10\}, \{25,15,30,15,8\} \} \\ \text{İş5} &= \{ \{7,11,7,20,14\}, \{17,36,3,13,25\} \} \\ \text{İş6} &= \{ \{33,14,24,26,6\}, \{9,18,4,16,23\} \} \\ \text{İş7} &= \{ \{22,15,8,24,20\}, \{2,40,11,18,28\} \} \\ \text{İş8} &= \{ \{30,13,10,4,34\}, \{18,33,35,17,40\} \} \\ \text{İş9} &= \{ \{28,38,37,21,19\}, \{26,38,25,15,3\} \} \\ \text{İş10} &= \{ \{8,30,19,9,18\}, \{33,38,6,6,12\} \} \end{aligned}$$

IBM ILOG CPLEX kullanılarak, ele alınan problemin optimum çözümü en büyük üretim süresi (Cmax) amaç fonksiyonu için 28 olarak bulunmuştur. Önerilen algoritma C# programlama dilinde kodlandıktan sonra aynı probleme uygulanmış ve optimum çözümle aynı sonuç, 28 bulunmuştur. Önerilen algoritmanın performansı küçük problem boyutları için test edildikten sonra nispeten daha büyük problemler ele alınmış ve algoritmanın davranışı gözlemlenmiştir.

Tablo 1. Farklı problem boyutları için önerilen algoritma sonuçları.

İş Sayısı x Aşama Sayısı x Makine Sayısı	Algoritma Performansı	
	Sonuç (Cmax)	İşlem Süresi (Sn)
40x5x5	294	33
40x5x10	56	35
40x10x5	633	93
40x10x10	123	99
40x15x5	956	184
40x15x10	201	192
50x5x5	440	46
50x5x10	99	50
50x10x5	936	135
50x10x10	175	141
50x15x5	1334	265
50x15x10	258	274
100x5x5	1152	157
100x5x10	289	163
100x10x5	2362	298
100x10x10	586	305
100x15x5	3576	588
100x15x10	954	600
250x5x5	3694	569
250x5x10	1316	578
250x10x5	7361	830
250x10x10	2708	865
250x15x5	11303	1636
250x15x10	4211	1670
1000x5x5	17243	4245
1000x5x10	7885	4288
1000x10x5	35003	13297
1000x10x10	15744	12496
1000x15x5	52127	51476
1000x15x10	23378	51546

Deney sonuçlarına bakıldığında çizelgelenen iş sayısı ve üretim yapılan aşama sayısı arttıkça Cmax değerlerinin arttığı, aşamalarda işlem yapan makine sayısının artması ile ise Cmax değerlerinin düştüğü gözlemlenmiştir. Algoritma yüksek iş sayıları ve üretim aşamalarına kadar zorlanmış ve algoritmanın verdiği sonuçların tutarlı olduğu gözlemlenmiştir.

SONUÇ VE ÖNERİLER (DISCUSSION AND SUGGESTIONS)

Bu çalışmada melez akış tipi çizelgeleme problemi için bir genetik algoritma tasarlanmıştır. Tasarlanan algoritmanın kromozom yapısı farklı işlem aşamaları için farklı iş permutasyonlarından oluşmaktadır. Algoritmanın bir çözüm adayının uygunluk değeri bulunurken o çözüm adayının farklı aşamalara göre oluşturduğu iş sırası göz önüne alınmış ve işler bu sıraya göre işlem aşamalarındaki makinelerde gezdirilerek o çözüm adayının uygunluk değeri hesaplanmıştır. Algoritma tasarımı gerçekleştirilirken algoritmanın farklı problem boyutlarına uyarlanabilir bir yapıda olmasına dikkat edilmiş, algoritmanın genetik operatörler (çaprazlama, mutasyon) uygulamalarını hızlı ve efektif bir şekilde gerçekleştirmesine imkan veren yaklaşımlardan (kromozom yapısının iş permutasyonu şeklinde tasarlanması, her bir genin ondalıklı sayılarla ifade edilmesi) yararlanılmıştır.

Algoritma, her bir çözüm adayının uygunluk değerini hesaplarken ele alınan problem için tasarlanan sistemi temel almaktadır. Çözüm adaylarının uygunluk değerleri, çözüm permutasyonundaki işlerin işlem birimlerinde gezdirilmesi ve seçilen amaç fonksiyonu (Cmax) doğrultusunda hesaplanmaktadır. Her bir çözüm adayı için uygunluk değerinin bu şekilde belirlenmesi hesap yoğun ve karmaşık bir işlem yükünü gerektirmektedir. Algoritmanın tutarlı sonuçlar vermesi birincil olarak uygunluk belirleme aşamasındaki işlemlerin doğruluğuna bağlıdır.

Çözüm popülasyonunun zaman içinde daha iyi çözümlere evrilmesi amaçlanmıştır. Bu bağlamda çaprazlama operatörünün tasarımı algoritmanın daha iyi çözümlere evrilmesini sağlamak açısından kritik bir öneme sahiptir. Anne ve babanın çocuk birey oluşumuna eşit oranda etki etmesi ile popülasyondaki iyi bireylerin özelliklerinin bir sonraki nesile aktarılabilmesi düşünülmüştür.

Yerel optimuma takılmanın önüne geçilebilmesi için popülasyonun sonraki nesillerde farklılaşması gerekmektedir. Mutasyon operatörü ile popülasyonun farklılaşması gerçekleştirilir. Algoritmada her gen bir ondalıklı sayı ile temsil edilmekte, mutasyon olasılığının çok küçük bir değer seçilmesine karşın her gen değerine her gen için farklı rassal sayılarla üretilen mutasyon olasılığı eklenir. Böylece mutasyon sonrası oluşturulan bireyin önceki bireyden farklı olma ihtimali artırılır.

Ele alınan problemin bir çizelgeleme problemi olması ve iş sayısına bir sınır getirilmemesi düşünüldüğü için algoritmanın hızlı bir şekilde iyi çözümlere evrilmesinin sağlanması gerekmektedir. Uygunluk tabanlı strateji ile ebeveyn ve çocuklardan oluşan çözüm havuzunun en iyileri sonraki

nesile aktararak hızlı bir şekilde daha iyi çözümlere ulaşılmasının sağlanması amaçlanmıştır.

Algoritma öncelikle nispeten küçük boyutlu bir probleme uygulanmış bulunan sonuç aynı problem için CPLEX solverin ürettiği optimum sonuç ile karşılaştırılmıştır. Algoritmanın küçük problem boyutları için iyi sonuç verdiği gözlemlendikten sonra algortima davranışı büyük boyutlu problemlere uygulanmıştır. Algoritmanın performansını büyük problem boyutlarında da kıyaslamak için çeşitli çözümler kullanılmış ancak problemin np zor yapısı gereği kıyaslama için sonuç elde edilmekte zorlanılmıştır. Mevcut çözümler ya çok uzun zamanda optimum çözüme ulaşmakta ya da karma tam sayılı programlama gereği tam sayı çözüm bulmakta zorlanmaktadır. Bu doğrultuda algoritmanın büyük problem boyutları için ürettiği sonuçlar kendi içinde gözlemlenmiştir.

Problem boyutu ve aşama sayısı arttıkça algoritmanın verdiği sonuç değerlerinin (Cmax) arttığı, makine sayısı arttıkça ise sonuç değerlerinin düştüğü görülmüştür. Yine aynı doğrultuda iş sayısı ve aşama sayısı arttıkça işlem sürelerinin arttığı görülmüş, algoritmanın farklı problem boyutları için tutarlı çalıştığına karar verilmiştir.

Bu çalışmada melez akış tipi çizelgeleme problemi için bir genetik algoritma önerilmiştir. Önerilen algoritmanın performansı çeşitli test problemleri ile ölçülmeye çalışılmıştır. Önerilen algoritma çizelgeleme literatüründeki bir çok farklı probleme uyarlanabilir şekilde tasarlanmıştır. İleriki çalışmalarda test kümesinin genişletilerek, algoritmanın farklı çizelgeleme problemlerine uygulanması ile performansını iyileştirecek yöntemlerin denenmesi amaçlanmaktadır.

KAYNAKÇA (REFERENCES)

- Allahverdi, A., Al-Anzi, F.S. (2006). Scheduling multi-stage parallel-processor services to minimize average response time. *Journal of the Operational Research Society* 57 (1), 101–110.
- Alaykyran K, Engin O, Doyen A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *Int J Adv Manuf Technol* 35(5–6), 541–50.
- Allaoui H, Artiba A. (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Comput Oper Res* ;33(5), 1399–419.
- Alisantoso D, Khoo LP, Jiang PY. (2003). An immune algorithm approach to the scheduling of a flexible PCB flow shop. *Int J Adv Manuf Technol*, 22(11–12), 819–27.
- Akrami B, Karimi B, Hosseini SMM. (2006). Two metaheuristic methods for the common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: the finite horizon case. *Appl Math Comput*.183(1), 634–45.
- Belkadi K, Gourgand M, Benyettou M. (2006). Parallel genetic algorithms with migration for the hybrid flow shop scheduling problem. *J Appl Math Decis Sci*, 1–17.
- Brockmann K, Dangelmaier W, Holthofer N. (1997). Parallel branch & bound algorithm for makespan optimal scheduling in flow shops with multiple processor. SOR'97.
- Cheng J, Yoshiyuki K, Kise H. (2001). A shifting bottleneck approach for parallel-machine flowshop scheduling problem. *J Oper Res Soc Jpn*, 44(2), 141–154.
- Carlier J, Neron E. (2003). An exact method for solving the multi-processor flowshop. *Rairo Rech Operationnelle Oper Res*, 34(1), 1–25.
- Dessouky MM, Dessouky MI, Verma SK. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *Eur J Oper Res* , 109(3), 620–31.
- Engin O, Doyen A. (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Comput Syst*, 20(6), 1083–95.

- Garcia JM, Lozano S. (2005). Production and delivery scheduling problem with time windows. *Comput Ind Eng* , 48:733–42.
- Guinet AGP, Solomon MM. (1996). Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *Int J Prod Res* , 34(6), 1643–54.
- Guinet A, Solomon MM, Kedia PK, Dussauchoy A. (1996). A computational study of heuristics for two-stage flexible flowshops. *Int J Prod Res*, 34(5), 1399–1415.
- Guirchoun S, Martineau P, Billaut J. (2005). Total completion time minimization in a computer system with a server and two parallel processors. *Comput Oper Res.*, 32(3):599–611.
- Gupta JND, Kruger K., Lauff V., Werner F., Sotskov Y.N. (2002). Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Comput Oper Res* , 29(10), 1417–39.
- Gupta JND, Tunc EA. (1998). Minimizing tardy jobs in a two-stage hybrid flowshop. *Int J Prod Res* , 36(9), 2397–417.
- Gupta JND, Hariri AMA, Potts CN. (1997). Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Ann Oper Res*, 69:171–91.
- Gupta JND, Sexton RS, Tunc EA. (2000). Selecting scheduling heuristics using neural networks. *Inform J Comput*, 12(2), 150–62.
- He LM, Sun SJ, Luo RZ. (2007). A hybrid two-stage flowshop scheduling problem. *Asia Pac J Oper Res*, 24(1),45–56.
- Haouari M, M'Hallah R. (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. *Oper Res Lett*, 21(1), 43–53
- Haouari M, Hidri L, Gharbi A. (2006). Optimal scheduling of a two-stage hybrid flow shop. *Math Methods Oper Res.*, 64(1), 107–24.
- Jin ZH, Ohno K, Ito T, Elmaghraby SE. (2002). Scheduling hybrid flowshops in printed circuit board assembly lines. *Prod Oper Manag.*, 11(2), 216–30.
- Jin ZH, Yang Z, Ito T. (2006). Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *Int J Prod Econ.*, 100(2), 322–334.
- Janiak A, Kozan E, Lichtenstein M, Oguz C. (2007). Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. *Int J Prod Econ*, 105(2), 407–24.

- Kurz ME, Askin RG. (2003). Comparing scheduling rules for flexible flow lines. *Int J Prod Econ.*, 85(3), 371–88.
- Kurz ME, Askin RG. (2004). Scheduling flexible flow lines with sequence-dependent setup times. *Eur J Oper Res.*, 159(1), 66–82
- Kim JS, Kang SH, Lee SM. (1997). Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. *Omega Int J Manag Sci.*, 25(5), 547–55.
- Li SL., (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. *Eur J Oper Res* ,102(1), 142–56.
- Logendran R, deSzoek P, Barnard F., (2006). Sequence-dependent group scheduling problems in flexible flow shops. *Int J Prod Econ* , 102(1), 66–86.
- Moursli O, Pochet Y., (2000). A branch-and-bound algorithm for the hybrid flowshop. *Int J Prod Econ.*, 64(1–3), 113–25.
- Nowicki E, Smutnicki C., (1998). The flow shop with parallel machines: a tabu search approach. *Eur J Oper Res* , 106(2–3), 226–53.
- Nawaz M, Ensore Jr EE, Ham I., (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91–5.
- Oguz C, Ercan MF, Cheng TCE, Fung YF., (2003). Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop. *Eur J Oper Res.*, 149(2), 390–403.
- Oguz C, Fung YF, Ercan MF, Qi XT., (2003). Parallel genetic algorithm for a flow-shop problem with multiprocessor Task. *ICCSA 2003, Lecture notes in computer sciences*, vol. 2667. p. 987–97.
- Oguz C, Ercan M., (2005). A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *J Scheduling*, 8(4), 323–51.
- Oguz C, Ercan MF., (1997). Scheduling multiprocessor tasks in a two-stage flow-shop environment. *Comput Ind Eng* , 33(1–2), 269–72.
- Ruiz, R., Vázquez-Rodríguez, J. A., (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1-18.
- Sawik T., (2000). Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Math Comput Model*, 31(13), 39–52.

- Soewandi H, Elmaghraby S., (2001). Sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Trans* , 33(11), 985–94.
- Sawik T., (2005). Integer programming approach to production scheduling for make- to-order manufacturing. *Math Comput Model*, 41(1), 99–118.
- Sawik T., (2006). Hierarchical approach to production scheduling in make-to-order assembly. *Int J Prod Res*, 44(4), 801–30.
- Suresh RK, Mohanasundaram KM., (2004). Pareto archived simulated annealing for permutation flow shop scheduling with multiple objectives. Proceedings of IEEE Conference on Cyber- netics and Intelligent Systems (CIS), vol 2. Singapore, December 1–3, pp 712–717
- Thornton HW, Hunsucker JL., (2004). A new heuristic for minimal makespan in flow shops with multiple processors and no intermediate storage. *Eur J Oper Res* ,152, 96–114.
- Tavakkoli-Moghaddam R, Safaei N, Sassani F., (2009). A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Comput Oper Res* , 36(2), 402–14.
- Wardono B, Fathi Y.,2004. “A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities”. *Eur J Oper Res* ,155(2):380–401.
- Wang H, Jacob V, Rolland E., (2003). Design of efficient hybrid neural networks for flexible flow shop scheduling. *Expert Syst* , 20(4), 208–31.
- Vandevonder, S., Ballestin, F., Demeulemeester, E., Herroelen, W., (2007). Heuristic procedures for reactive project scheduling. *Computers & Industrial Engineering*, 52(1), 11-28.
- Ying KC, Lin SW., (2006). Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach. *Int J Prod Res* , 44(16), 3161–3177.
- Zandieh M, Fatemi Ghomi SMT, Moattar Hussein SM., (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Appl Math Comput* ,180(1), 111–27.
- Zhang W, Yin CY, Liu JY, Linn RJ., (2005). Multi-job lot streaming to minimize the mean completion time in hybrid flowshops. *Int J Prod Econ.*, 96(2), 189–200.

EXTENDED ABSTRACT

In this study, hybrid flow shop scheduling problem which have many application areas in both production and service sectors is discussed. In order to solve the problem, a genetic algorithm was designed and the performance of the algorithm analyzed for the various problems sizes. Empirical results indicate that proposed algorithm offer promising results.

INTRODUCTION

When many production and service systems are analyzed operations are made ready after a series of processing units. Often these operations follow the same sequence. The processing environment which the processing units are ordered in series and the operations follow this sequence called as flow shop. If there are more than one processor units at least one stage, then this processing environment is called as a hybrid flow shop. Exact algorithms or heuristics can be applied to solve the hybrid flow shop scheduling problem. The proposed algorithm for the hybrid flow shop scheduling problem in this work is a new genetic algorithm. The important feature of the proposed genetic algorithm is its chromosome structure. A solution candidate is composed of operation permutations of each processing stages.

The proposed algorithm handles each stage separately, but operations are scheduled considering the combined system.

PROPOSED GENETIC ALGORITHM

Normally jobs are waiting to process for each stage, in hybrid flow shop environment. Therefore in the proposed algorithm, a chromosome composed of job permutations of stages. For example, a five job two stage hybrid flow shop problem solution candidate is $\{\{J2, J1, J5, J4, J3\}, \{J1, J3, J2, J5, J4\}\}$. Initial population is generated randomly and random key representation is choosed for gene representation to take advantage of the floating point numbers in recombinations.

In the proposed genetic algorithm a solution candidate fitness value is calculated by simulating the designed system. Firstly job permutation is generated for each stages, and then jobs are given the fastest idle processing units according to this permutation order. This procedure is applied all jobs in all stages, and fitness value of this solution candidate is computed by taking biggest completion time of the whole stages.

To increase the randomness tournament selection method is choosed as a parent selection method. By taking the weighted averages of the genes

crossover is done for each parent couple, and swap mutation is applied to create new childs.

Fitness based survivor selection mechanism is choosed to decide next generation. The next generation is generated by using a solution pool is composed of combination of parents and childs. The solutions in this pool ordered according to their fitness value and fitter ones are choosed as a next generation.

DISCUSSION AND SUGGESTIONS

In this study, a genetic algorithm is designed for the hybrid flow shop scheduling problem. The designed algorithm chromosome structure is composed of various permutations for different stages. While calculating fitness values of a solution candidates, job sequences of permutations are considered. The jobs are travelled in the process units according to the permutation sequence to calculate fitness values. The algorithm is designed to be a generic structure that can be adapted to various problem sizes.

The algorithm is firstly applied to a relatively small size problems. The algorithm performance is compared with optimum result. The algorithm results for small size problems very close to the optimum results. After testing algorithm performance for small size problems, algorithm is applied bigger problems. Hybrid flow shop problem Np-Hard nature prevent reaching optimum result for bigger problems using available solvers. Therefore, the performance of the algorithm for bigger problem sizes is evaluated by observing the behaviour of the algorithm for different operation and stage numbers. The algorithm gives also consistent results for bigger problem sizes.

In this study, the proposed algorithm performance is measured with various test problems. However, the proposed algorithm is designed to be adapted to many different problems in the scheduling literature. The algorithm performance can be tested with different problems from the scheduling literature. The new method and genetic operators also can be developed to increase the performance of the algorithm for different problems.