



Paper Produced from PhD Thesis Presented at
Graduate School of Natural and Applied Sciences, Yıldız Technical University
Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi Kapsamında Hazırlanan Yayın



Research Article / Araştırma Makalesi

JOB SCHEDULING WITH THE HELP OF DOMINANCE PROPERTIES AND GENETIC ALGORITHM ON HYBRID FLOW SHOP PROBLEM

Pelin ALCAN*¹, Hüseyin BAŞLIGİL²

¹*Istanbul Gelişim Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Müh. Bölümü, Avcılar-İSTANBUL*

²*Yıldız Teknik Üniversitesi, Makine Fakültesi, Endüstri Mühendisliği Bölümü, Yıldız-İSTANBUL*

Received/Geliş: 18.12.2014 Accepted/Kabul: 03.11.2015

ABSTRACT

Products are accessed easier with the industrial revolution. Product accessibility increases the customer demand. Consequently, to satisfy increased customer demands companies expand their manufacturing capacities. After a literature review, we determined that there is hardly any study on “unrelated parallel machine and set up time constrained Hybrid Flow Shop” problems. Specifically, the techniques, i.e. dominance properties, that help heuristics methods are not used. In almost all studies, either heuristics or meta-heuristics methods are applied. The problem complexity plays an important role in selecting the solution methodologies. In this dissertation, genetic algorithm, which is an evolutionary algorithm, with dominance property is used to solve the proposed problem.

Keywords: Hybrid flow shop, scheduling, heuristic, metaheuristic, genetic algorithm, dominance properties.

HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİNDE ÜSTÜNLÜK ÖZELLİKLERİ VE GENETİK ALGORİTMA YARDIMI İLE İŞ SIRALAMA

ÖZ

Sanayi devrimi ile ürünlerin daha kolay ulaşılabilir olması, müşterilerin taleplerini arttırmış, artan müşteri taleplerini karşılamak için de işletmeler, üretim kapasitelerini arttırmaya başlamışlardır. Literatür araştırması sonucunda “Hibrit Akış Tipi, ilişkisiz paralel makineli ve hazırlık zamanı kısıtlı” sistemler üzerinde yapılan çalışmaların az olduğu, hele “üstünlük özellikleri” gibi sezgisellerin işini kolaylaştıran yöntemlerin çok da kullanılmadığı görülmüştür. Makalelerin çoğunda sezgisel ve meta sezgisel yöntemler denenmiştir. Problemlerin oldukça zor olması da yöntem seçimlerinde etkili rol oynamaktadır. Bu tez çalışması kapsamında sunulan problemin çözümünde, evrimsel algoritmalarından genetik algoritma, üstünlük özellikleri konusu ile birlikte denenmiştir.

Anahtar Sözcükler: Hibrit akış tipi, çizelgeleme, sezgisel, metasezgisel, genetik algoritma, üstünlük özellikleri.

1. GİRİŞ

Hibrit Akış Tipi (Hybrid Flow Shops-HFS) ise, m aşamalı seri bir sistemde, bir grup işin olduğu, genel bir üretim sistemidir. HFS sistemi, bir dizi aşamadan ve her aşamada da bir ya da daha çok paralel makinelerden oluşmaktadır. HFS probleminin “standart” formunda bütün işler

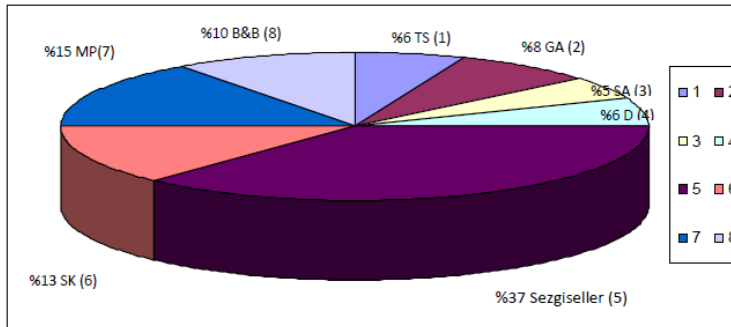
* Corresponding Author/Sorumlu Yazar: e-mail/e-ileti: palcan@gelisim.edu.tr, tel: (505) 453 60 23

ve makineler zamanın sıfır anında hazırdirler. Verilen bir aşamadaki makineler özdeştir. Herhangi bir makine bir zamanda sadece bir işi işleme tabi tutabilmekte ve herhangi bir iş, belli bir zamanda bir makine ile işleme girebilmektedir. Hazırlık zamanları ihmal edilebilir. Aşamalar arasında bulunan ara stok kapasiteleri sınırsızdır. Problem bilgisi deterministik yani kesindir ve önceden bilinmektedir.

Bu çalışmada, HFS yapısı içerisinde, her aşamada, literatürde bu tip sistemler içinde oldukça az yer bulan “ilişkisiz paralel makineler” kullanılmıştır. Yapılan modellemede görülmüştür ki, örnek boyutu büyüdükçe model yavaşlamaktadır. Daha sonra modelin çözümüne yardımcı olması açısından “sezgisel” teknikler denenmiştir. Sezgisel tekniklerin en çok kullanılanlarından birisi olan “genetik algoritma” yöntemi uygulanmıştır.

Genetik algoritmanın kullanılabilirliğinin ve etkinliğinin daha da artırılması amacıyla “üstünlük/baskınlık özellikleri” kullanılmıştır. Verilen bir sıralama işlemi için “baskınlık özellikleri (Dominance Properties-DP)” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Büyük boyutlu problemlerin çözümü zor olmaktadır. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabilir.

Linn ve Zhang (1999) [1] çalışmalarında, Hibrit Akış Tipi (HFS) ’nin genel bir çizimini yapmış ve gelecek araştırmaları için yönlendirmelerde bulunmuşlardır. Genoulaz (2000) [2] maksimum gecikmeyi minimize etmek için “zaman aralığı” ve “öncelik kısıtları” ile HFS problemini incelemiştir. Bunun için, “problem bazlı” sezgiseller önerilmiştir. Oğuz vd. (2003) [3] çok işlemcili, 2 aşamalı HFS probleminde tamamlanma zamanını minimize etmek için “sezgisel algoritmalar” önermişlerdir. Pek çok çalışma (sıralama problemleri üzerinde) ya hazırlık zamanlarını görmemezlikten gelmekte ya da iş sıralamasının olduğu her makinenin bağımsız olduğunu varsaymaktadır. Bunun sonucunda, Zandieh vd. (2006) [4] çalışmalarında “sıralama” bağımlı hazırlık zamanlarının olduğu bir HFS problemini (SDST) incelemiştir. Sezgisel olarak “immün (immün) algoritma (IA)” düşünülmüştür. Low vd. (2008) [5] ilişkisiz makineler ile 2 aşamalı HFS problemini incelemiştir. Amaç, sistemde tamamlanma zamanını minimize etmektir. Kullanılan sezgiseller, modifiye edilmiş Johnson kuralı ve dağıtım kurallarıdır. Behnamian vd. (2010) [6] ise HFS ortamında, sıralama hazırlık zamanları ile işleri sıralamış ve hibrit bir sezgisel kullanmışlardır. Uygulamada ise, “değişken komşuluk yaklaşımı”, ACO (Karınca kolonisi Optimizasyonu) ve benzetimli tavlama yöntemleri uygulanmıştır. Wang ve Liu (2013) [7] 2 aşamalı beklemez HFS (benzer paralel makinelili) probleminde, Genetik Algoritma’yı kullanmışlardır. Marichelvam vd. (2014) [8] çalışmalarında, tamamlanma zamanını minimize etmeyi amaç eden, çok aşamalı bir HFS sıralama probleminde, son zamanlarda geliştirilen “cuckoo araştırması (ICS)” metasezgiselini denemişlerdir.



Şekil 1. B&B (Dal-Sınır), TS (Tabu arama), GA (Genetik Alg.), SA (Benzetimli Tavlama), MP (Matematiksel Programlama), SK (Sevkiyat/Dağıtım Kuralları), Sezgiseller ve D. (Diğer yöntemler) in HFS makalelerinde dağılımı

Şekil 1’de, 2012 senesine kadar olan HFS makalelerinde, Dal-Sınır, TS (Tabu arama), GA (Genetik Alg.), SA (Benzetimli Tavlama), MP (Matematiksel Programlama), SK (Sevkiyat/Dağıtım Kuralları), Sezgiseller ve Diğer yöntemlerin oranları görülmektedir.

Tezin ilerleyen bölümleri şu şekilde organize edilmiştir: Bölüm 2’de Hibrit Akış Tipi Çizelgeleme kısmı anlatılmıştır. 3. Bölüm’de Genetik Algoritma, 4. Bölüm’de Kurulum Zamanlı Sıralama Problemleri İçin Üstünlük Özellikleri (DP) Yaratımı, 5. Bölüm’de Genetik Algoritma ve Üstünlük Özellikleri Kullanımı Uygulaması irdelenirken, 6. Bölüm’de sonuçlar ve öneriler anlatılmıştır.

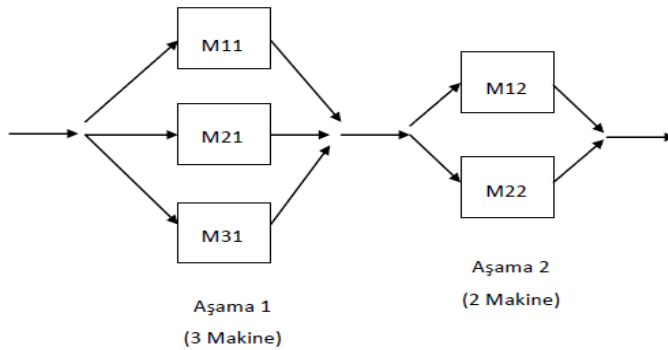
2. HİBRİT AKIŞ TİPİ ÇİZELGELEME (HFS)

Her kademe için birden çok paralel makineler ile akış tipi zaman çizelgelemesinin oluşturulması durumu genel olarak “hibrit akış tipi (HFS) atölye sıralama” problemi olarak adlandırılmaktadır. Bu durum birçok gerçek dünya uygulamasında karşımıza çıkar [9].

Hibrit Akış Tipi, genel olarak, verilen bir amaç fonksiyonunu en iyileyen seri m aşamalı durumda prosese giren bir dizi n işin sıralanması durumudur. Farklı işler aşamaları aynı sırada ziyaret ederler. Her aşamada, bir iş sadece bir makine tarafından işlem görür [10]. Şekil 2’de bir HFS sistemi görülmektedir.

Bir dizi değişken ise şu şekildedir;

1. Proses aşamalarının sayısı en azından 2’dir.
2. Her kademe en az bir kademe $M(k) > 1$ olmak üzere $M(k) \geq 1$ paralel makine içerir.
3. Bütün işler takip eden aynı üretim aşamasında (aşama 1, aşama 2,...aşama m) işlem görmektedir. İş herhangi bir kademeyi en azından birinde işleme tabi tutulmak şartı ile geçebilir.
4. Aşama k’ da, her j işi, bir p_{jk} proses zamanına sahiptir. Biz k aşamasında j işinin işlenmesini O_{jk} operasyonu ile göstermeliyiz.



Şekil 2. HFS yapısı

2.1 İlişkisiz Paralel Makine ile Problem Tanımı

Esnek atölye akış tipi sistemi bir dizi k proses aşamalarından oluşan bir diziyi $O = \{1, \dots, t, \dots, k\}$ tanımlar. Her aşama t dir. Ve $t \in O$ ’dur. m^t ilişkisiz paralel makinelerinden oluşan dizi ise şöyledir; $Mt = \{1, \dots, i, \dots, mt\}$. n bağımsız işin dizisi $J = \{1, \dots, j, \dots, n\}$, M_1, \dots, M_k ile tanımlanan bir makine dizisi üzerinde prosese girmektedir. Her iş j ve $j \in J$ ’dir. Bırakış zamanı (release date) r_j ’dir ve $r_j \geq 0$ ’dir. Ayrıca teslim tarihidi $d_j \geq 0$ ’dir. Her t, $t \in O$ aşaması için standart bir proses zamanı bulunmaktadır. İlişkisiz paralel makinelerin kabulü ile, t aşamasında i makinesindeki j işinin p_{ij}^t proses zamanı şöyledir; ps_j^t/v_{ij}^t ’dir. Burada, ps_j^t , t aşamasındaki j işinin standart proses zamanıdır. v_{ij}^t ise, t aşamasında, i makinesindeki j işinin nispi hızıdır [11].

İşlerin proses kısıtlamaları şöyledir;

1. herhangi bir makinede işler öne almasız prosese girmektedir.
2. belli bir zamanda her makinede sadece bir operasyon işlem görmektedir.
3. bir işin operasyonları aşamalar arasında, sırası ile bindirme olmaksızın işlem görmelidir.
4. işlerin bölünmesine izin verilmez.

Paralel şekilde m sayıda birbirinden farklı makine vardır. j işi i makinesinde v_{ij} hızı ile işlem görebilmekte ve işlem süresi p_j/v_{ij} olarak gerçekleşmektedir. Makine hızlarının tüm işlerden bağımsız olması durumunda makine hızı v_{ij} yerine v_i gelmektedir.

3. GENETİK ALGORİTMA

Genetik Algoritma temel olarak, evrimsel sistemin doğal işleyişini canlandırabilecek şekilde biçimlendirilmiştir. GA da tüm sezgisel yaklaşımlar gibi “rastlantı” aramaya dayanmaktadır [12]. Standart bir GA yordamı ise şöyle olmaktadır [13]:

- Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Toplumda, yani popülasyonda bulunacak birey sayısı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Birey sayısı oluşturulur ve ardından probleme bağlı olarak kromozomların kodlanması gerekir.

- Toplumdaki her kromozomun ne kadar iyi olduğu bulunur. Bu amaçla kullanılan fonksiyona “uygunluk fonksiyonu (fitness function)” denir. Bu fonksiyon yardımı ile kromozomların uygunluklarının bulunmasına ise “evrimleşme” denilir. Uygunluk fonksiyonu, GA’ nın beynidir. GA’ da probleme özel çalışan tek fonksiyon, uygunluk fonksiyonudur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek, onların bir bakıma şifresini çözmekte (decoding) ve sonra bu parametrelere göre hesaplamayı yaparak, kromozomların uygunluğunu bulmaktadır.

- Seçilen kromozomları eşleyerek “yeniden kopyalama” ve “değiştirme operatörleri” uygulanır. Sonuçta yeni bir toplum oluşturulur. Bu eşleme kromozomların uygunluk değerlerine göre yapılır. Bu seçim için, rulet tekerleği seçimi, turnuva seçimi gibi seçme yöntemleri bulunmaktadır. Yeniden kopyalama, genlerdeki genetik bilginin birinden diğerine geçmesi işlemine benzediği için çaprazlama olarak adlandırılır. Bu işlem toplumda çeşitliliği sağlar. Değiştirme yani mutasyonun etkisi sadece bir çözüm üzerinde olmaktadır.

- Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.

- Tüm kromozomların uygunlukları tekrar hesaplanır ve yeni toplumun başarısı bulunur.

- İşlemler tekrarlanarak verilmiş zaman içinde daha iyi olan yeni nesillerin oluşturulması sağlanır.

- Sonuç olarak, toplumların hesaplanması sırasında en iyi bireyler bulunduğu çözüm elde edilmiş olur.

3.1. Genetik Algoritma Operatörleri

Holland tarafından 1975’de geliştirilen Genetik Algoritma (GA), pek çok kombinatoriyel optimizasyon problemleri içerisinde başarılı olarak uygulanmıştır. Genetik algoritma rassal arama tekniklerini kullanarak çözüm bulmaya çalışan, “parametre kodlama” esasına dayalı bir çözüm arama tekniğidir [14].

Genetik Algoritmalar, yeni bir nesil oluşturabilmek için 3 aşamadan geçmektedirler [14]:

- Eski nesildeki her bir bireyin uygunluk değerini hesaplamak.
- Bireyleri, uygunluk değerini göz önüne alarak (uygunluk fonksiyonu) seçmek.
- Seçilen bireyleri, çaprazlama (crossover), mutasyon (mutation) gibi genetik operatörler kullanarak eşleştirmek.

Genetik algoritmanın etkinliği kullanılan parametrelere bağlı olarak değişmektedir [15]. Akış tipi çizelgeleme problemlerinin genetik algoritma ile çözümünde çözüm kalitesi üzerinde etkisi olduğu bilinen bazı parametreler kullanılarak deneyler yapılmıştır. Çaprazlama ve mutasyon, Genetik Algoritma yönteminin en önemli kısımlarıdır. Yöntemin performansı bu iki operatörden ciddi manada etkilenmektedir. Bu operatörlerin her ikisi de, bir popülasyondan yeni bir popülasyon oluştururken kullanılmaktadırlar.

3.1.1. Uygunluk Fonksiyonu

Uygunluk fonksiyonu tanım olarak, kromozomların çözümde gösterdikleri başarı derecesini belirleyen bir değerlendirme işlevidir. Hangi kromozomların bir sonraki nesle aktarılacağına ve hangi kromozomların yok olacağına, uygunluk değerlerinin büyüklüğü ile karar verilir. Zaten Genetik Algoritma’ da kullanılan değerlendirme işlevi/uygunluk fonksiyonu, problemin amaç işlevini oluşturmaktadır. Uygunluk fonksiyonun formül olarak gösterimi ise genel olarak şöyledir [16].

Tüm kısıtlar için bir popülasyondaki ceza puanları toplamı:

$$\sum_{k=0}^t \sum_{i=0}^n \sum_{j=0}^l P_k * C_{ij} \quad (1)$$

Uygunluk fonksiyonu:

$$(f) = \frac{1}{1 + (\sum_{k=0}^t \sum_{i=0}^n \sum_{j=0}^l P_k * C_{ij})} \quad (2)$$

l= kromozomdaki gen sayısı,

n= popülasyondaki kromozom sayısı,

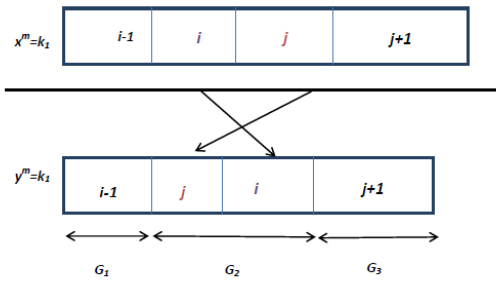
t= kısıt sayısı,

C_{ij} = i. Kromozomdaki ceza puanına sahip j. gen,

P_k = k. kısıta ait ceza puanı.

4. KURULUM ZAMANLI SIRALAMA PROBLEMLERİ İÇİN ÜSTÜNLÜK ÖZELLİKLERİ (DP) YARATIMI

Chang ve Chen (2011) [17] makalelerinde, “genetik algoritma” gibi meta sezgisellerin sıralama problemlerine uyumunun yavaş olmasından bahsederler. Bu uyumu geliştirmek ve ilerletmek için ise, üstünlük özelliklerinin (dominance properties-DP) meta sezgisellere eklenmesi önemli olmuştur. Optimal sıralamanın üstünlük özellikleri, iki bitişik işin, i ve j’nin yer değişimi temeline dayanmaktadır. Üstünlük özellikleri, meta-sezgiseller uygulanmadan önce, “ilk” verimli çözümleri elde ederler. Böylelikle, DP “ilk iyi” çözümleri ürettiği zaman, meta-sezgiseller optimuma daha hızlı yakınsayabilecektir. “Üstünlük Özellikleri”, Genetik Algoritma ve benzeri sezgisel yöntemlere, çözüme “daha hızlı” yakınsaması adına yardım etmektedir. Şekil 3’de paralel makineler üzerindeki iş değişimleri



Şekil 3. Bir paralel makine üzerindeki iş değişimleri

Notasyonlar şöyle olacaktır;

$[j]$: $[j]$ konumundaki iştir.

$P_{[j][k]}$: $[k]$ makinesinde $[j]$ pozisyonunda ki işin proses zamanıdır.

$S_{[i][j][k]}$: k makinesinde, i işinden sonra j pozisyonundaki işin kurulum zamanı.

$AP_{[i][j][k]}$: k makinesinde i işinden sonra j pozisyonundaki işin düzeltilmiş proses zamanı (adjusted).

Böylece $AP_{[i][j][k]} = P_{[j][k]} + S_{[i][j][k]}$ 'dir.

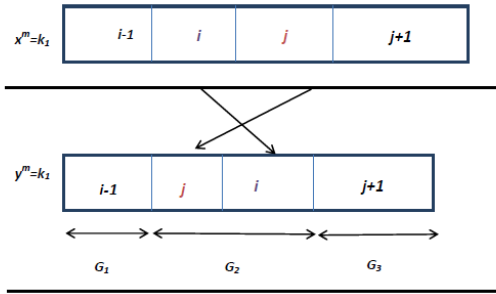
C_{k1} : k_1 'deki tamamlanma zamanı.

G_{1k} : k makinesinde i işinden önceki iş dizisi.

G_{2k} : k makinesinde $[i]$ işi ve $[j+1]$ işleri arasındaki iş dizisi.

G_{3k} : k makinesinde $[i]$ işinden sonraki iş dizisi.

Daha önce de belirtildiği gibi, iş değişimleri ile alakalı 2 şart bulunmaktadır; bunlar makineler arası ve makineler içindeki değişimlerdir.



Şekil 3. Bir paralel makine üzerindeki iş değişimleri

Notasyonlar şöyle olacaktır;

$[j]$: $[j]$ konumundaki iştir.

$P_{[j][k]}$: $[k]$ makinesinde $[j]$ pozisyonunda ki işin proses zamanıdır.

$S_{[i][j][k]}$: k makinesinde, i işinden sonra j pozisyonundaki işin kurulum zamanı.

$AP_{[i][j][k]}$: k makinesinde i işinden sonra j pozisyonundaki işin düzeltilmiş proses zamanı (adjusted).

Böylece $AP_{[i][j][k]} = P_{[j][k]} + S_{[i][j][k]}$ 'dir.

C_{k1} : k_1 'deki tamamlanma zamanı.

G_{1k} : k makinesinde i işinden önceki iş dizisi.

G_{2k} : k makinesinde $[i]$ işi ve $[j+1]$ işleri arasındaki iş dizisi.

G_{3k} : k makinesinde $[i]$ işinden sonraki iş dizisi.

Daha önce de belirtildiği gibi, iş değişimleri ile alakalı 2 şart bulunmaktadır; bunlar makineler arası ve makineler içindeki değişimlerdir.

4.1. Makineler İçi Değişim

Makineler arası değişim ile ilgili olarak düşünülen 2 durum söz konusudur; bitişik değişim (adjacent exchange) ve bitişik olmayan değişim (non- adjacent exchange). Aşağıdaki formülasyon "adjacent exchange" ile ilgili olmaktadır:

Bitişik Olan Değişim

Aşağıdaki şart sağlanıyorsa, değiştirilen çizelge asıl çizelgeden daha iyidir;

$$(AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i][k]} - AP_{[j][j][k]}) + (AP_{[i][j+1][k]} - AP_{[j][j+1][k]}) < 0$$

Bitişik Olmayan Değişim

Aşağıdaki durumlar sağlanırsa, i işi ve j işi yer değiştirecektir.

$$(AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i+1][k]} - AP_{[j][i+1][k]}) + (AP_{[j-1][i][k]} - AP_{[j-1][j][k]}) + (AP_{[j][j+1][k]} - AP_{[j][j+1][k]}) < 0$$

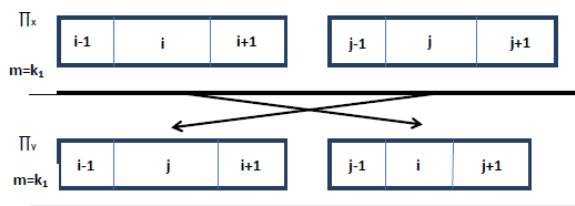
$$G_{1[k]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k]} = G'_{1[k]}$$

$$G_{2[k]} = G_{1[k]} + AP_{[i-1][i][k]} + AP_{[i][i+1][k]} + AP_{[j-1][j][k]} + AP_{[j][j+1][k]}$$

$$G_{3[k]} = G_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$

$$G'_{2[k]} = G_{1[k]} + AP_{[i-1][j][k]} + AP_{[j][i+1][k]} + AP_{[j-1][i][k]} + AP_{[i][j+1][k]}$$

$$G'_{3[k]} = G'_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$



Şekil 4. Bitişik olmayan işlerin değişimi

Δ farklılık değeri ise şöyledir;

$$\Delta = \Pi_y - \Pi_x = G_{2[k]} - G_{2[k]} = (AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i+1][k]} - AP_{[j][i+1][k]}) + (AP_{[j-1][i][k]} - AP_{[j-1][j][k]}) + (AP_{[j][j+1][k]} - AP_{[j][j+1][k]})$$

Δ sıfırdan küçük ise yer değiştirme yapılmaktadır.

4.2. Makineler Arası Değişim

Makineler arası değişim (Intra-machine exchanging) kısmı, her hangi iki farklı makinede bulunan i ve j işlerinin birbiri arasındaki değişimini tartışmaktadır. Paralel makine sayıları 2'ye eşit ya da daha fazladır, k1 ve k2 gösterimleri i işi ve j işi için makine sayılarını göstermektedir.

$$\text{Max} \{ (G_{3[k_1]} + (AP_{[i-1][j][k_1]} - AP_{[i-1][i][k_1]}) + (AP_{[j][i+1][k_1]} - AP_{[j][i+1][k_1]}), G_{3[k_2]} + (AP_{[i-1][j][k_2]} - AP_{[i-1][i][k_2]}) + (AP_{[j][i+1][k_2]} - AP_{[j][i+1][k_2]}) \} < C_{\text{max}}$$

k₁' in amaç farklılığı

k₁'deki her iş dizisinin tamamlanma zamanı şöyledir;

$$G_{1[k_1]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k_1]} = G'_{1[k_1]}$$

$$G_{2[k_1]} = G_{1[k_1]} + AP_{[i-1][i][k_1]} + AP_{[i][i+1][k_1]}$$

$$G_{3[k_1]} = G_{2[k_1]} + \sum_{a=j+2}^n AP_{[a-1][a][k_1]}$$

$$G'_{2[k_1]} = G'_{1[k_1]} + AP_{[i-1][j][k_1]} + AP_{[j][i+1][k_1]}$$

$$G'_{3[k_1]} = G'_{2[k_1]} + \sum_{a=j+2}^n AP_{[a-1][a][k_1]}$$

Buna göre, makine k1 in amaç farklılığı şöyledir;

$$\Delta_{k1} = \Pi'_x - \Pi_x = (AP_{[i-1][j][k1]} - AP_{[i-1][i][k1]}) + (AP_{[j][i+1][k1]} - AP_{[i][i+1][k1]})$$

k₂' nin amaç farklılığı

Makine k₂' deki tamamlanma zamanı şöyledir;

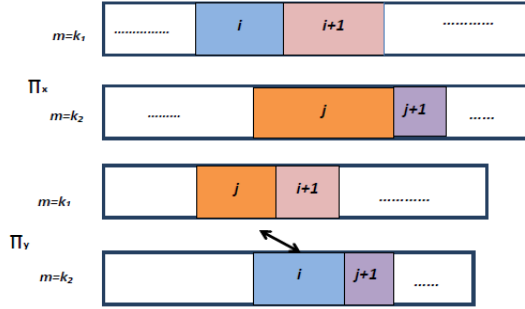
$$G_{1[k_2]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k_2]} = G'_{1[k_2]}$$

$$G_{2[k_2]} = G_{1[k_2]} + AP_{[j-1][j][k_2]} + AP_{[j][j+1][k_2]}$$

$$G_{3[k_2]} = G_{2[k_2]} + \sum_{a=j+2}^n AP_{[a-1][a][k_2]}$$

$$G'_{2[k_2]} = G'_{1[k_2]} + AP_{[j-1][i][k_2]} + AP_{[i][j+1][k_2]}$$

$$G'_{3[k_2]} = G'_{2[k_2]} + \sum_{a=j+2}^n AP_{[a-1][a][k_2]}$$



Şekil 5. Farklı makinelerdeki i ve j işlerinin yer değişimi

Amaç farklılığını, Π'_y 'yi Π_y 'den çıkararak bulabiliriz.

$$\Delta_{k1} = \Pi'_y - \Pi_y = (AP_{[j-1][i][k2]} - AP_{[j-1][j][k2]}) + (AP_{[i][j+1][k2]} - AP_{[j][j+1][k2]})$$

Sonuç olarak, değişen sıralamanın C'_{max} değeri, $\text{Max}(G_{3[k1]} + \Delta_{k1}, G_{3[k2]} + \Delta_{k2})$ olmaktadır.

Bu nedenle, eğer C'_{max} , C_{max} 'dan daha az ise, i ve j işi yer değiştirir.

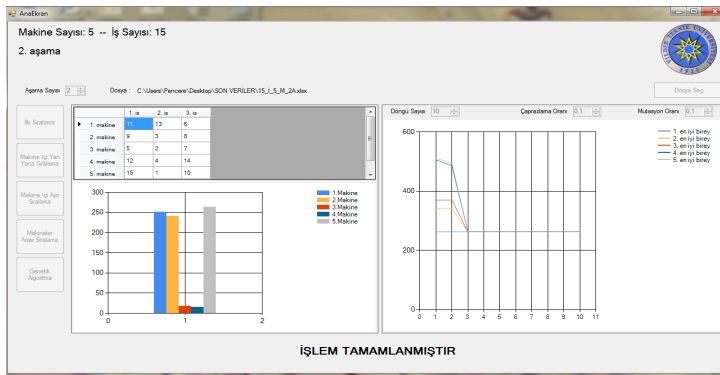
5. GENETİK ALGORİTMA VE ÜSTÜNLÜK ÖZELLİKLERİ KULLANIMI UYGULAMASI

Model, OPL programlama dilinde uzun zaman harcamaya başladıktan sonra, bir sezgisel/metasezgisel yöntemin kullanılmasının iyi olacağına karar verilmiştir. Sezgiselin çözümü ve kodlanması için, programlama dili olarak "C# (Csharp) Programlama Dili" kullanılmıştır. Önceki kısımlarda da söylendiği gibi, genetik algoritma kullanımı içerisinde parametre seçimleri önemli olmaktadır. Kullanılan parametreler Çizelge'de gösterilmiştir.

Çizelge 1. Kullanılan Parametreler

Çaprazlama oranı	0.1
Mutasyon oranı	0.1
Döngü sayısı	10
Başlangıç popülasyon sayısı	30
Mutasyon şekli	Keyfi üç iş değiştirme
Çaprazlama şekli	Pozisyona dayalı çaprazlama
Seçim Kuralı	Turnuva seçimi

Örneğin, 15/5/2 örneği OPL programında çok uzun zaman harcarken, C# ile çözülmeye çalışılacaktır. Bu verilere göre oluşturulan Csharp ekran çıktısı Şekil 6'daki gibi olmaktadır. Bu sonuç, genetik algoritma ile baskınlık özellikleri birleştirilerek alınmıştır. Diğer sonuçlar ise, Çizelge 2'deki gibidir.



Şekil 6. 15 İş 2 Aşama 5 Makine için Csharp ekran çıktısı

Çizelge 2. Örnek Veriler Tablosu

ÖRNEK VERİLER	GA-DP Cmax	OPL Cmax
4 iş / 2 aşama / 2 makine	206	205
6 iş / 3 aşama / 3 makine	175	152
8 iş / 4 aşama / 4 makine	287	285
9 iş / 3 aşama / 3 makine	245	Çözemedi
10 iş / 3 aşama / 5 makine	236	Çözemedi
15 iş / 2 aşama / 5makine	267	Çözemedi

6. SONUÇLAR VE ÖNERİLER

Bu çalışma içerisinde, öncelikle “iş sıralama” problemleri içerisinde daha az ilgililenen bir konu olan “Hibrit Akış Tipi (HFS)” sistemler incelenmiş ve bu konu üzerinde bir model düşünülmüştür. HFS yapısı içerisinde her aşamada ise, literatürde bu tip sistemler içinde oldukça az yer bulan “ilişkisiz paralel makineler” kullanılmıştır. Yapılan modelleme prosesinde görülmüştür ki, örnek boyutu büyüdükçe, model yavaşlamaktadır ve çözümü de zorlaşmaktadır. Sıralama problemleri zaten doğası gereği oldukça zor yapıda olan ve çözümü de komplike bir yapı arz eden problemlerdir. Bu nedenle, modellemenin ardından modelin çözümüne yardımcı olması açısından, C sharp programı üzerinde yazılıp kodlanan “metasezgisel” bir teknik denenmiştir. Çizelgeleme Literatüründe, Metasezgisel çalışmalar içerisinde en çok kullanılan

yöntemlerden birisi olan “genetik algoritma” uygulanmıştır. Verilen bir sıralama işlemi için “baskınlık özellikleri” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Daha önce de söylendiği gibi, büyük boyutlu problemlerin çözümü zor olmaktadır. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabileceği bu çalışma içerisinde verilen en temel ve en önemli noktalardan birisi olmaktadır.

Sonuçta, Genetik algoritmanın kullanılabilirliği ile etkinliğinin daha da artırılması adına, “üstünlük/baskınlık özellikleri” metodolojiye eklenerek, oldukça başarılı sonuçlar elde edilmiştir. İlerleyen senelerde, çalışmayı geliştirmek adına, farklı meta sezgisel ya da sezgisel yöntemlerle “baskınlık özellikleri” birleştirilebilir. Ya da HFS sistemi içerisinde baskınlık özelliklerinin eklendiği ve bu şekilde denenen bir meta sezgisel yöntem içerisinde, farklı amaç fonksiyonları ele alınabilir. Ayrıca diğer meta sezgisel yöntemler de bu tip bir HFS sistemine dahil edilebilir.

REFERENCES / KAYNAKLAR

- [1] Linn R., Zhang W., Hybrid Flow Shop Scheduling: A Survey, *Computers & Industrial Engineering*, 37, (1999) 57-61.
- [2] Genoulaz V.B., Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness, *Int. J. Production Economics*, 64, (2000) 101-111.
- [3] Oguz C., Ercan M.F., Cheng T.C.E., Fung Y.F., Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop, *European Journal of Operational Research*, 149, (2003) 390-403.
- [4] Zandieh M., Fatemi Ghomi S.M.T., Moattar Hussein S.M., An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times, *Applied Mathematics and Computation*, 180, (2006) 111-127.
- [5] Low C.Y., Hsu C.J., Su C.T., A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines, *Computers & Operations Research*, 35, 3, (2008) 845-853.
- [6] Behnamian J., Fatemi Ghomi S.M.T., Zandieh M., Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flowshop with sequence dependent setup times, *International Journal of Production Research*, 48, 5, (2010). 1415-1438.
- [7] Wang S., Liu M., A heuristic method for two-stage hybrid flow shop with dedicated machines, *Computers & Operations Research*, 40, 1, (2013) 438-450.
- [8] Marichelvam M.K., Prabaharan T., Yang X.S., Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan, *Applied Soft Computing*, 19, (2014) 93-101.
- [9] Ruiz R., Rodríguez J.A.V., The hybrid flow shop scheduling problem, *European Journal of Operational Research*, 205, (2010) 1-18.
- [10] Belkadi K., Gourgand M., Benyettou M., Parallel Genetic Algorithms With Migration For The Hybrid Flow Shop Scheduling Problem, *Journal of Applied Mathematics and Decision Sciences*, (2006), ID 65746: 1-17.
- [11] Jungwattanakit J., Reodech M., Chaovalitwongse P., Werner F., A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria, *Computers & Operations Research*, 36, 2, (2009) 358-378.
- [12] Cura T., Modern Sezgisel Teknikler ve Uygulamaları, *Papatya Yayıncılık*, (2008).
- [13] Nabyev V.V., Yapay Zeka, *Seçkin Yayıncılık*, 2.Baskı, Ankara, (2005).
- [14] Goldberg D.E., Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley Publishing Company, USA, (1989).

- [15] Chen C.L., Vempati V.S., Aljaber, N., An Application of Genetic Algorithms for Flowshop Problems, *European Journal of Operational Research*, 80, (1995) 389-396.
- [16] Birođlu S., Güvenç U., Genetik Algoritma Çözümü Gerçekleştiren Atölye Çizelgeleme Probleminde Ürün Sayısının Etkisi, Gazi Üniversitesi Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, Beşevler, Ankara, (2004).
- [17] Chang P.C., Chen, S.H., Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times, *Appl. Soft Comput.*, 11, 1, (2011) 1263-1274.