

Bir boyutlu kutulama probleminin eniyilenmesi için hiper-sezgisel paralel bir algoritma

Tansel DÖKEROĞLU*

Türk Hava Kurumu Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara

Geliş Tarihi (Received Date): 22.02.2016

Kabul Tarihi (Accepted Date): 20.12.2016

Özet

Bir boyutlu kutulama problemi (1BKP), endüstri mühendisliğinin üzerinde en çok çalışılan NP-Zor kombinatoriyal problemlerinden bir tanesidir. Büyük sayıda (elliden fazla) parça içeren problem kümeleri için en iyi çözümün bulunması klasik kaba kuvvet algoritmaları ile yüz yıllarca sürebilmektedir. Bu yüzden (yaklaşık)-optimal çözümleri ile eniyilemeyi tam olarak ya da düşük performans kayıpları ile kısa sürelerde bulabilen sezgisel algoritmalar sıklıkla kullanılmaktadır. Bu çalışma ile birlikte, Gruplama Genetik Algoritmalarında (GGA) kullanılan sezgisel kutulama tekniklerinden sadece bir tanesini kullanan klasik yaklaşımlar yerine, aynı anda birçok sezgisel kutulama tekniğini kullanan hiper-sezgisel paralel bir algoritma (HPGG-1BKP) geliştirildi. En Uygun Boşluğu Doldur (EUBD), İlk Bulduğun Boşluğu Doldur (İBBD) ve En Küçük Boşluğu Bırakarak Doldur (EKBBDD) sezgisel kutu doldurma algoritmaları bu algoritmada aynı anda paralel olarak kullanıldı. 1228 benchmark problemi üzerinde yapılan deneyler sonucunda %88.1 başarı ile 1070 optimal sonuç elde edildi. Geri kalan problemler için de sadece bir kutu daha fazla kullanan çözümler üretilerek sonuçlar eniyilendi. Önerilen algoritma Falkenauer GGA ile karşılaştırıldığında %9'a varan iyileşmeler elde edildi.

Anahtar kelimeler: Genetik, bir boyutlu kutulama, paralel, hiper-sezgisel

A parallel hyper-heuristic algorithm for the optimization of one-dimensional bin packing problem

Abstract

One-Dimensional Bin Packing Problem (1DBPP) is one of the most well-known NP-Hard problems of industrial engineering. The execution time of a brute force approach

* Tansel DÖKEROĞLU, tdokeroglu@thk.edu.tr, <http://orcid.org/0000-0003-1665-5928>

algorithm for finding the optimal solution for its problem instances with several items (more than 50) can spend more than hundreds of years. Therefore, heuristic algorithms that can find (near)-optimal solutions are preferred with their reasonable optimization times. With this study, we propose a novel parallel hyper-heuristic Grouping Genetic Algorithm (HHPGGA-1DBPP) that uses several heuristics concurrently, whereas classical Grouping Genetic Algorithms (GGA) use only a single one during the optimization. The solutions of the proposed algorithm outperform the classical ones'. Best Fit Decreasing (BFD), First Fit Decreasing (FFD), and Minimum Bin Slack (MBS) are the bin-oriented heuristics used in the proposed algorithm. With the experiments carried out on 1,228 problem instances, 1,070 of the problems (88.1%) are solved optimally. The remaining problems are optimized by producing only a single extra bin. These experimental results show that the proposed algorithm can outperform Falkenauer GGA. The obtained results are improved up to 9%.

Keywords: Genetic, one-dimensional bin packing, parallel, hyper-heuristic.

1. Giriş

Bir Boyutlu Kutulama Problemi (1BKP), mühendislik, lojistik ve üretim gibi birçok alanda karşımıza esas problem ya da başka bir problemin parçası olarak sıkça çıkan NP-Zor kombinatoriyal bir eniyileme problemidir [1-3]. Bu problemde, kapasitesi c olan kutular için n adet parça en az sayıda kutu ile paketlenmeye çalışılır. Kullanılabilecek kutuların sayısı sınırsız olarak kabul edilir ve kapasiteleri, $c > 0$ 'dır. Parça $i \in \{1, \dots, n\}$ ve boyutu $s_i > 0$ iken 1BKP Denklem 1'deki gibi ifade edilmektedir.

$$\forall k: \sum_{i \in \text{bin}(k)} s_i \leq c \quad (1)$$

1BKP'nin amacı n adet parçayı en az sayıda kutu (M) içerisine paketleyebilmektir (Bkz: Denklem 2).

$$M \geq \left\lceil \sum_{i=1}^n s_i / c \right\rceil \quad (2)$$

1BKP'nin çözümü için daha önce birçok sezgisel algoritma geliştirilmiş olmasına rağmen, bu çalışma ile ilk defa olarak hiper-sezgisel paralel bir algoritma önerilmiştir. Hiper-sezgisel algoritmalar, birçok sezgisel eniyileme yöntemini aynı anda kullanabilen ve hepsinin farklı özelliklerinden aynı anda faydalanabilen etkin bir yöntemdir [4-6]. Sezgisel yaklaşımlar problemin özelliğine göre farklı performanslar gösterebilmektedir. Bu yüzden, hepsinin en kuvvetli taraflarından faydalanmayı amaçlayan bu yaklaşım, tek başına çalışan sezgisel algoritmalara göre daha iyi sonuçlar üretebilmektedir. Çalışmamıza ilham kaynağı olan Wolpert ve Macready tarafından geliştirilen "No Free Lunch" (NFL) adlı teoride [7] bu konuyla ilgili detaylı bilgi sunulmuştur.

Sezgisel algoritmalar, yerel en iyi çözümlere takılmaları engellenebilip ne kadar fazla deneme yapabilirlerse optimal değerlere o kadar daha fazla yaklaşıma şansı bulmaktadırlar. Bu süreç sırasında birçok alternatif çözümü denemeleri ve bu çözümlerin uygunluklarını görmeleri gerekmektedir. Paralel hesaplama araçları bu

anlamda çok daha fazla deneme yapabilme imkânı vermekte ve eklenen işlemci sayısına bağlı olarak performans artışı sağlayabilmektedir. Önerdiğimiz algoritmada bu yüzden paralel bir geliştirme ortamı olan MPI kullanıldı. Her işlemcide ayrı olarak tutulan ve birbirinden farklı olan popülasyonların nesiller boyu çaprazlama ve mutasyon işlemleri gerçekleştirilerek en iyi çözümlere ulaşması sağlandı. Bu işlem belirtilen nesil sayısı kadar tekrar edildi. Önerilen algoritma ile paralel hiper-sezgisel bir Gruplama Genetik Algoritmasının (GGA ilk örneklerinden birisi sunulmuş olmaktadır [8].

GGA'da kullanılan sezgisel kutulama tekniklerinden sadece bir tanesini kullanan yaklaşımların yerine, aynı anda birçok sezgisel kutulama tekniği kullanan hiper-sezgisel paralel bir algoritma (HPGG-1BKP) bu çalışmada ilk defa olarak geliştirilmiş oldu. En Uygun Boşluğu Doldur (EUBD), İlk Bulduğun Boşluğu Doldur (İBBD) ve En Küçük Boşluğu Bırakarak Doldur (EKBBDD) sezgisel kutu doldurma algoritmaları aynı anda paralel olarak farklı işlemciler üzerinde kullanıldı. 1228 benchmark problemi üzerinde yapılan deneyler sonucunda %88.1 başarı ile 1070 optimal sonuç elde edildiği görülmüştür. Geri kalan problemler için sadece bir kutu daha fazla bulan çözümler üretilerek sonuçlar eniyilendi. Önerilen algoritmanın diğer güncel GGA yaklaşımlara göre daha etkin olduğu deneysel olarak gösterilmiştir.

Makalenin ikinci bölümünde; 1BKP için geliştirilen güncel sezgisel algoritmalar ile ilgili bilgi verilmektedir. Üçüncü bölümde; önerilen hiper-sezgisel paralel algoritmanın bileşenleri hakkında bilgi verildi. Dördüncü bölümde; deney ortamı, benchmark problemleri ve elde edilen sonuçların diğer algoritmalar ile karşılaştırılarak değerlendirilmesi yapıldı. Son bölümde; önerilen algoritmanın genel bir değerlendirmesi ve gelecekte yapılabilecek çalışmalar ile ilgili bilgi verildi.

2. Önceki çalışmalar

Bu bölümde 1BKP için geliştirilmiş güncel sezgisel algoritmalar hakkında bilgi sunulmaktadır. Martello ve Toth'un kitabında kutulama problemleri hakkında çok detaylı olarak bilgi verilmektedir [9]. En Uygun Boşluğu Doldur (EUBD) ve İlk Bulduğun Boşluğu Doldur (İBBD) algoritmaları en çok bilinen ve kullanılan sezgisel algoritmalarlardır. Her iki algoritma da tüm parçaları büyükten küçüğe göre sıraladıktan sonra, en uygun olan ya da ilk buldukları kutuya parçaları yerleştirirler. Parça için uygun bir yer bulunamazsa yeni bir kutu eklenir [2]. En Küçük Boşluğu Bırakarak Doldur (EKBBDD) algoritması, Gupta ve Ho tarafından 1999 yılında önerilmiştir [10]. EKBBDD' algoritması da EKBBDD algoritmasının parça azaltma teknikleri (İng. reduction) kullanılarak geliştirilmiş bir sürümüdür. Fleszar ve Charalambous 2011 yılında kutulanmamış parçaların ortalama ağırlıklarını dikkate alarak yeni kutuları dolduran sezgisel bir algoritma olan (sufficient average weight) SAW'yi önerdiler [11]. İki ve üç boyutlu kutulama problemleri için sezgisel memetik algoritmaların da geliştirildiği görülmektedir [6].

Falkenauer 1994 yılında yaptığı çalışmalar ile ilk olarak GGA için etkin bir kromozom yapısı geliştirdi ve dejenerasyonun önüne geçerek 1BKP'ni yüksek bir başarı oranı ile çözdü [12]. Bu çalışmada, Falkenauer GGA'larda EUBD sezgisel yaklaşımını kullanmayı tavsiye etmektedir. Dokeroglu ve Cosar yeni melez sezgisel kutulama algoritmaları geliştirerek paralel ortamda 1BKP'ni iyi çözen yöntemler buldular [2]. İki ve üç boyutlu kutulama problemleri için paralel ve bazı hiper-sezgisel algoritmalar

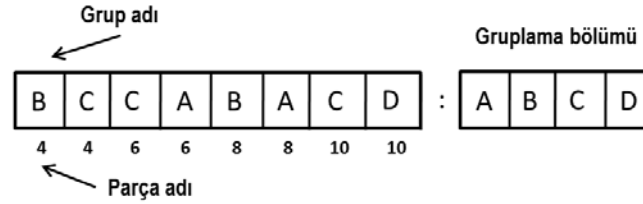
geliştirilmiş olduğu bilinmesine rağmen, bir boyutlu kutulama problemleri için bizim önerdiğimiz gibi hiper-sezgisel paralel bir algoritmanın geliştirilmemiş olduğu görülmektedir [13].

3. Önerilen algoritma

Bu bölümde, önerilen hiper-sezgisel paralel algoritmanın (HPGG-1BKP) bileşenleri ve tekrar yerleştirme sezgisel algoritması olarak kullanılan EUBD, İBBD ve EKBBBD hakkında bilgi verilmektedir.

3.1. Kromozom yapısı

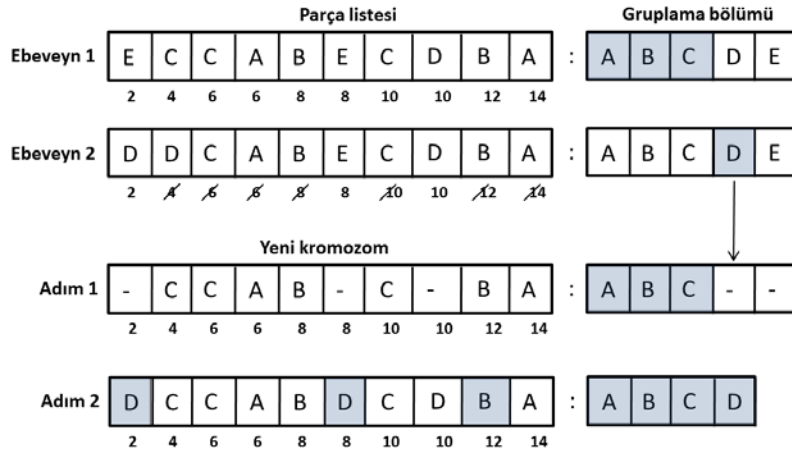
GGA’da en önemli problemlerden biri, çaprazlama ve mutasyon sırasında kromozomların elde ettikleri bilgi seviyesini dejenere olan yeni kromozomlardan dolayı kaybetmeleridir. Bu problem klasik Holland tipi kromozom yapısı ile çalışan GGA’larda sıkça karşılaşılan bir problemidir. Bu yüzden Falkenauer, yeni bir kromozom yapısı önermiş ve bu problemin daha iyi şekilde çözümüne katkıda bulunmuştur [12]. Önerdiğimiz algoritmada dejenerasyonu engelleyen ve Falkenauer tarafından geliştirilmiş olan kromozom yapısı kullanılmaktadır. Şekil 1’de kullanılan kromozom görülmektedir. Sol taraftaki bölüm parçaların hangi grupta olduğunu gösterirken sağ tarafta kullanılan bölüm ise grupların isimlerini göstermektedir. Sol taraftaki bölümün altındaki rakamlar parçaların boyutunu göstermektedir (kutu kapasitesi, $c=20$ olarak kabul edilmiştir). Çaprazlama, mutasyon ve ters çevirme işlemleri sağ taraftaki bölüm üzerinde yapılmaktadır. Parçaların değişen grup isimleri sol taraftaki bölüm üzerinden takip edilerek güncellenmektedir.



Şekil 1. Önerilen algoritmada kullanılan Falkenauer kromozom yapısı. Gruplama bölümü sağ tarafta yer almaktadır.

3.2. Çaprazlama işlemi

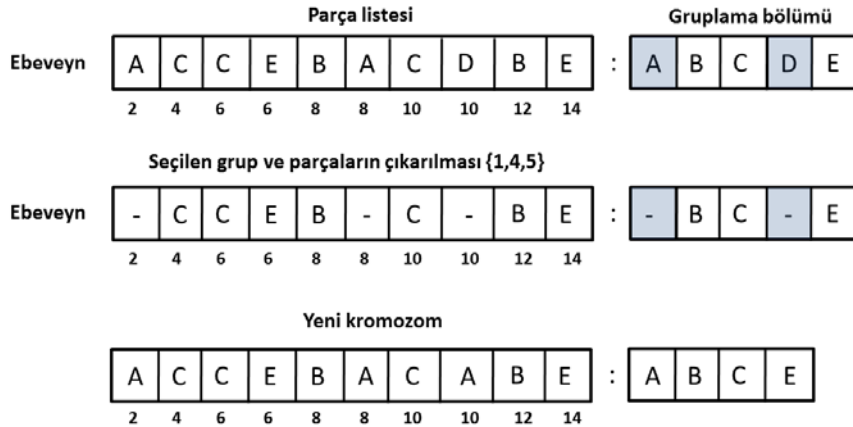
Önerilen algoritmada Şekil 2’de örneği verilmiş olan çaprazlama işlemi kullanılmaktadır. Çaprazlama işlemi sırasında popülasyondan iki adet ebeveyn kromozom seçilmektedir. Bu seçilen kromozomların gruplama bölümleri kesilerek yeni bir kromozom oluşturulmaktadır. Örnekte de görüldüğü gibi, ebeveyn 1’den A, B ve C grupları ebeveyn 2’den de D grubu alınarak yeni bir kromozom oluşturulmuştur. Çaprazlama işlemi sırasında bozulan kutulardan bazı parçalar artmakta ve bu parçalar yeni kutular oluşturularak ya da diğer kutulardaki uygun yerlere tekrar yerleştirilmektedir. Bu işlem sırasında sezgisel algoritmalar kullanılmaktadır. Bu sezgisel algoritmaların etkinliği algoritmanın tüm performansını etkilemektedir. Örnekte 5’er adet kutu ile paketleme yapan kromozomlardan 4 kutulu yeni bir kromozom oluşturulmuştur. Falkenauer bu tekrar yerleştirme işlemi için İBBD algoritmasını tavsiye etmektedir. Biz yaptığımız çalışmada, birçok sezgisel algoritmayı aynı anda paralel olarak deneyerek daha iyi sonuçlar elde ettik.



Şekil 2. Kullanılan çaprazlama işlemi örneği

3.3. Mutasyon işlemi

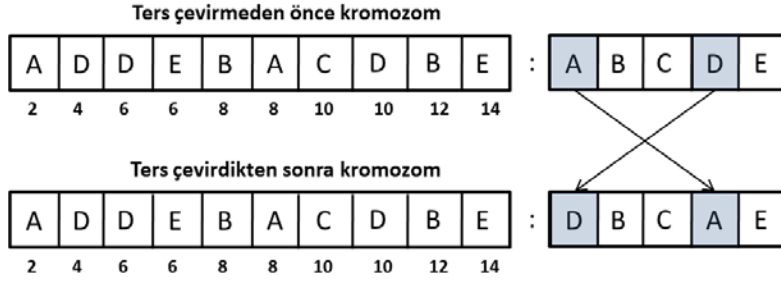
Mutasyon işlemi esnasında rasgele olarak bazı gruplar seçilmektedir. Bu seçilen gruplarda bulunan parçalar, kromozom dışına alınarak sezgisel algoritma ile geride kalan kutuların içerisine ya da yeni kutulara tekrar doldurulurlar. Mutasyon tek bir kromozom üzerinde çalışır. Şekil 3’de bir mutasyon örneği verilmiştir. Ebeveyndeki A ve D grupları seçilerek kromozomdan çıkarılmış ve ardından tekrar kromozoma yerleştirilmişlerdir. 5 kutu ile paketlenmiş olan ebeveyn, mutasyon sonrası 4 kutu ile paketlenebilir daha iyi bir çözüme evrilmektedir.



Şekil 3. Mutasyon işlemi örneği

3.4. Ters çevirme işlemi

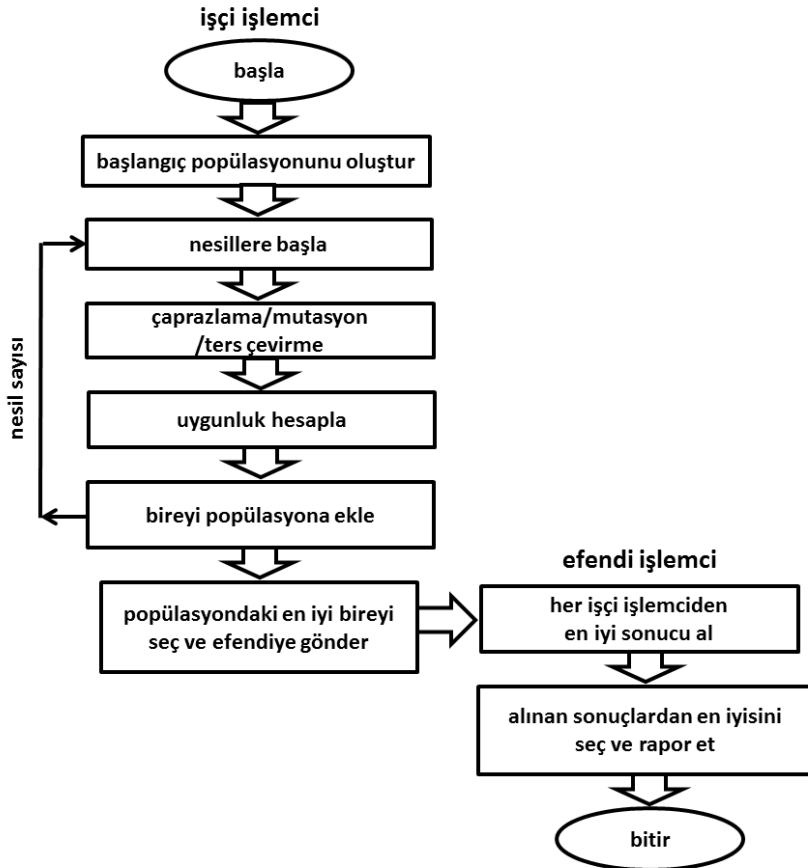
Ters çevirme işlemi grupların çaprazlama ve mutasyonlarda seçilebilme olasılığını çeşitlendirmek amacıyla kullanılan bir işlemdir. Kromozomun grup sayısı ile ilgili bir değişiklik yapmamaktadır. Bunun yanında çaprazlama ve mutasyon için değişik olasılıkların doğmasına sebep olmakta ve farklılaşmayı sağlamaktadır. Şekil 4’teki örnekte, A ve D gruplarının yerleri değiştirilerek bir ters çevirme işlemi yapılmıştır. Kromozomun sol tarafında herhangi bir değişiklik yapılmamıştır.



Şekil 4. Ters çevirme işlemi örneği

3.5. Önerilen algoritma, HPGG-1BKP

İşçi işlemciler kendi hafızaları üzerinde başlangıç popülasyonu oluşturarak belirlenen nesil sayısı kadar çaprazlama, mutasyon ve ters çevirme işlemleri ile ellerindeki çözümleri iyileştirirler. İşçi işlemciler daha önceden belirlenen sezgisel algoritmalara göre kutulara yeniden yerleşme işlemi yapar. Birinci işlemci EUBD, ikinci işlemci İBBD ve üçüncü işlemci de EKBBBD sezgisel algoritmasını kullanır. Diğer işlemciler de sayılarına göre bu dağılıma uyarlar. Her işlemcide farklı bir popülasyon oluşturularak problemin farklı çözüm alanlarının daha etkin olarak aranması sağlanmıştır. İşçi işlemciler nesilleri bitince, elde ettikleri en iyi çözümü efendi işlemciye gönderirler. Efendi işlemci, tüm işlemcilerden gelen çözümleri toplar ve en iyisini seçerek esas çözüm olarak belirler. Önerilen algoritmanın akış diyagramı Şekil 5'te sunulmuştur.



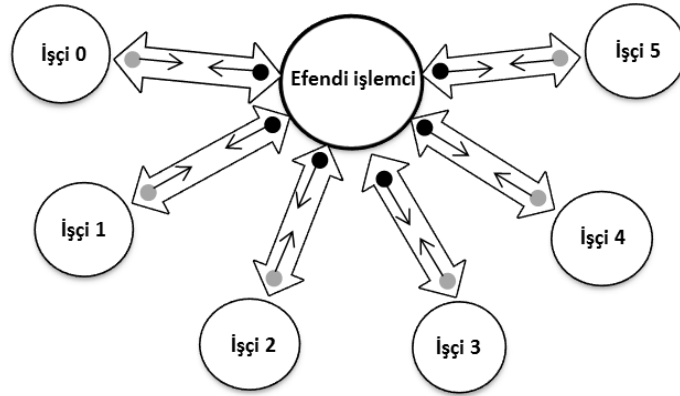
Şekil 5. Önerilen algoritmanın akış diyagramı

3.6. Kullanılan sezgisel algoritmalar

HPGG-1BKP algoritması her işlemcide farklı bir sezgisel algoritma kullanır. Parçalar kutulara işlemciye tanımlanmış olan sezgisel algoritmaya uygun olarak tekrar yerleştirilir. Biz bu çalışmada, EUBD, İBBD ve EKBBBD algoritmalarını kullandık. Falkenauer İBBD algoritmasını tavsiye etmiş olmasına rağmen diğer sezgisel algoritmaların farklı problemler örnekleri için daha iyi sonuç verebildikleri bu çalışma ile gösterilmiş olmaktadır. EUBD algoritması, parçaları büyükten küçüğe sıraladıktan sonra var olan kutuların içerisinde bulunan en uygun yere parçayı yerleştirir, eğer kutuların içerisinde yer bulamazsa yeni bir kutuya parçayı yerleştirir. İBBD algoritması da aynı sıralamayı yaptıktan sonra ilk bulduğu kutudaki boşluğa parçayı yerleştirir. Eğer kutularda uygun bir yer bulamazsa yeni bir kutu ekler. EKBBBD algoritması, her seferinde en yüksek oranda doluluğa sahip kutuyu oluşturmaya çalışır. Bütün alt kümeleri deneyerek en az boşluk bırakan alt kümeyi çözüm kümesi olarak tespit eder. Diğer ilk iki algoritmaya göre çalışma zamanı daha yüksektir.

3.7. Algoritmanın iletişim topolojisi

HPGG-1BKP algoritmasındaki işlemciler, efendi ve işçi işlemciler olmak üzere ikiye ayrılmışlardır. İşçi işlemciler ile efendi işlemci arasında sadece nesillerin bitmesinden sonra bir haberleşme olmaktadır. Bu yüzden dolayı haberleşme topolojisi oldukça ölçeklenebilir bir özelliğe sahiptir. Paralel uygulamalarda, (İng. master-slave) olarak bilinen ve birçok uygulamada etkin şekilde kullanılan bu topoloji, orta sayıda (50-1000 arasında) işlemcilerin bulunduğu yüksek performanslı bilgisayarlarda ölçeklenebilir bir iletişim ortamı sunmaktadır. Şekil 6'da, HPGG-1BKP algoritmasının işlemciler arasında gerçekleştirdiği iletişimin topolojisi görülmektedir. Bu konfigürasyonda tek efendi işlemci bulunmakta ve işçi işlemciler nesiller bitince ellerindeki en iyi çözümü efendi işlemciye göndermektedirler.



Şekil 6. Geliştirilen algoritma için bir efendi ve altı işçi işlemciden oluşan haberleşme topolojisi (İng. Master-slave topology).

4. Deney ortamı ve elde edilen sonuçların değerlendirilmesi

Deneyler, her birinde 2 CPU bulunan 46 düğümlü ve toplam 368 işlemcisi bulunan ODTÜ Bilgisayar Mühendisliği'nin yüksek performanslı bilgisayarı üzerinde gerçekleştirildi. Bu hesaplama ortamında toplamda 736 GB RAM kapasitesi bulunmakta olup, düğümler arasında iki adet 24 kapılı yüksek hızlı Gbps Ethernet anahtarları bulunmaktadır. Bu iletişim altyapısı, işlemciler arasından 8Gbps'lık bir haberleşme ortamı sağlamaktadır.

Deneylerde üç benchmark problem kümesi üzerinde çalışmalar yapıldı. set_1, set_2 [14] [15] ve hard28 [16]. Problem örnekleri ile ilgili detaylı bilgi Tablo 1’de sunulmuştur.

Tablo 1. Benchmark problem örnekleri hakkında detaylı bilgi.

Problemin Adı	Örnek sayısı	Parça Ağırlıkları	Kutu kapasitesi (c)	Parça sayısı (n)
set_1	720	[1,100]	{100, 120, 150}	{50, 100, 200, 500}
set_2	480	kutuda [3, 9] parça	1000	{50, 100, 200, 500}
hard_28	28	[1, 800]	1000	{160, 180, 200}

Tablo 2’de, geliştirilen HPGG-1BKP algoritması için kullanılan parametreler ve değerleri sunulmuştur. Bu değerler daha önceki çalışmalarımızdan en iyi değere yakın olduğunu bildiğimiz parametrelerdir [2]. Bu değerler ile GGA tüm işçi işlemcilerde çalıştırıldıktan sonra efendi işlemciye en iyi değer gönderilmekte ve en iyi sonuç elde edilmektedir.

Tablo 2. GGA için kullanılan parametreler ve değerleri

Parametrenin adı	Değeri
Popülasyonun boyu	40
Nesil sayısı	100
Transfer edilen mak. gen sayısı	%50
Transfer edilen min. gen sayısı	%10
Mutasyon oranı	%1
Ters çevirme oranı	%10

Önerilen algoritma, C++ ve MPI kütüphaneleri kullanılarak geliştirildi. Sonuçların güvenilirliğini artırmak için deneyler on defa tekrarlanarak ortalama sonuçlar alındı. Deney sonuçları önceki çalışmalar bölümünde bahsedilen, GGA-İBBD (Falkenauer Algoritması), EUBD, İBBD ve EKBBBD algoritmalarının elde ettikleri sonuçlar ile karşılaştırıldı. Geliştirilen HPGG-1BKP algoritması eklenen işlemci sayısına doğru orantılı olarak performansını artırmaktadır. Yapılan karşılaştırmalarda eşit bir ortam sağlamak için GGA-İBBD (Falkenauer) algoritması da 128 işlemci ile çalıştırılırken, deterministik olarak çalışan (her çalışmada aynı sonucu veren, değişken olmayan) ve çalışma süreleri oldukça düşük olan diğer algoritmalarda tek işlemci kullanıldı.

Tablo 3, 4 ve 5’te, 1228 adet benchmark problemi üzerinde yapılan deneylerin sonuçları ve diğer algoritmalar ile buldukları en iyi çözüm sayıları, fazla kutu sayıları ve çalışma zamanları karşılaştırmalı olarak sunulmaktadır. 1070 adet (%88.1) problem optimal olarak çözülürken sadece bir adet fazla olmak üzere toplamda 170 adet fazla kutu üretildiği gözlemlendi. HPGG-1BKP’ye en yakın başarıyı sağlayan Falkenauer’in GGA-İBBD algoritmasına göre set_1’de %9.1’lik, set_2’de %5.9’luk daha yüksek bir başarı sağlandı. EUBD ve İBBD algoritmaları çalışma zamanları diğerlerine göre daha hızlı olmasına rağmen, elde ettikleri optimal çözüm sayısı açısından HPGG-1BKP algoritmasının çok uzağında kaldılar. EUBD ve İBBD algoritmaları ile kıyaslandığında HPGG-1BKP algoritması sırasıyla set_1 için %43.0 ve %18.1 oranında, set_2 için ise %81.3 ve %72.8 oranlarında daha fazla optimal çözüm bulmaktadır. HPGG-1BKP algoritmasının çalışma zamanı daha fazla olmasına rağmen pratik olarak kabul edilebilir limitler içerisindedir.

Tablo 3. Set_1 problem kümesi için HPGG-1BKP algoritmasının diğer algoritmalar ile karşılaştırılması (128 işlemci ile)

Algoritma	Bulunan optimal çözüm sayısı	Fazla kutu sayısı	Çalışma zamanı (s)
EUBD	451	451	9.2
İBBD	546	283	9.3
GGA-İBBD (Falkenauer)	591	129	125.4
Önerilen (HPGG-1BKP)	645	75	128.3

Tablo 4. Set_2 problem kümesi için HPGG-1BKP algoritmasının diğer algoritmalar ile karşılaştırılması (128 işlemci ile)

Algoritma	Bulunan optimal çözüm sayısı	Fazla kutu sayısı	Çalışma zamanı (s)
EUBD	225	767	7.1
İBBD	236	748	7.0
GGA-İBBD (Falkenauer)	385	95	85.4
Önerilen (HPGG-1BKP)	408	72	84.3

Tablo 5. Hard28 problem kümesi için HPGG-1BKP algoritmasının diğer algoritmalar ile karşılaştırılması (128 işlemci ile)

Algoritma	Bulunan optimal çözüm sayısı	Fazla kutu sayısı	Çalışma zamanı (s)
EUBD	2	51	0.5
İBBD	5	23	0.5
GGA-İBBD (Falkenauer)	5	23	4.8
Önerilen (HPGG-1BKP)	5	23	5.2

Bu bölümde, önerilen algoritmanın hard28 problem kümesi için elde ettiği sonuçların literatürde bulunabilen EUBD, İBBD, EKBBBD, EKBBBD' ve SAWMBS' algoritmaları ile karşılaştırılması yapıldı. hard28 problem seti mevcut benchmarklar içerisinde en zor olanıdır. Bu yüzden algoritmanın bu problem seti üzerindeki performansı diğer algoritmalar ile karşılaştırıldığında okuyucuya iyi bir fikir verecektir.

Tablo 6. hard28 problem kümesi için elde edilen çözümler ve çalışma zamanlarının diğer algoritmalar ile karşılaştırılması

Algoritmanın adı	Bulunan optimal çözüm sayısı	Çalışma zamanı (ms)
EUBD	2	2.3
EKBBBD	2	4.2
EKBBBD'	3	3.6
İBBD	5	2.2
SAWMBS'	5	129.9
GGA-İBBD (Falkenauer)	5	5442.9
Önerilen (HPGG-1BKP)	5	5200.1

Sonuçlara bakıldığında optimal olarak bulunabilen çözüm sayısı 5'i geçen algoritma bulunmamaktadır. Önerdiğimiz algoritma da bunlardan bir tanesidir. Bu anlamda,

mevcut algoritmalar içerisinde en iyilerinden birisi olarak değerlendirilebilir. Sonuçların detayları Tablo 6’da sunulmuştur.

5. Sonuçlar ve tartışma

Bu çalışma ile 1BKP için ilk defa olarak hiper-sezgisel paralel bir eniyileme algoritması önerilmiştir. Önerilen HPGG-1BKP algoritması ile her işlemci üzerinde oluşturulan çözüm popülasyonları farklı sezgisel algoritmalar ile aynı anda eniyilenerek mevcut sezgisel çözüm algoritmalarından daha iyi sonuçlar elde edilmiştir. Deneyler esnasında 1228 benchmark problemi ile çalışılmıştır. Problemlerin 1070 tanesi (%88.1) optimal olarak bulunmuş ve geri kalan çözümlerde sadece bir adet fazla kutu kullanılmıştır. GGA Falkenauer algoritması ile karşılaştırıldığında %9 oranında iyileşme sağlanmıştır.

Gelecekte yapılacak çalışmalarda daha yüksek sayıda işlemcinin bulunduğu bulut bilişim ortamlarına uyumlu ölçeklenebilir ve farklı haberleşme topolojileri öneren yeni hiper-sezgisel algoritmalar ile iki ve üç boyutlu kutulama problemlerinin eniyilenmesi planlanmaktadır.

Kaynaklar

- [1] Garey, M. R., and Johnson, D. S., Computers and intractability: **A guide to the theory of NP-completeness**. W.H. Freeman, (1979).
- [2] Dokeroglu, T., and Cosar, A., Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. **Computers and Industrial Engineering**, 75, 176-186, (2014).
- [3] Stawowy, A., Evolutionary based heuristic for bin packing problem. **Computers and Industrial Engineering**, 55, 465–474, (2008).
- [4] Burke , E.K., McCollum, B., Meisels, A., Petrovic, S., and Qu, R., A graph-based hyper-heuristic for educational timetabling problems, **European Journal of Operations Research** 176, 1, 177–192, (2007).
- [5] Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., and Woodward, J.R., A classification of hyper-heuristic approaches, in: **Handbook of Metaheuristics**, Springer US, 449–468, (2010).
- [6] Beyaz, M., Dokeroglu, T., and Cosar, A. Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems. **Applied Soft Computing**, 36, 236-245, (2015).
- [7] Wolpert, D.H., and Macready, W.G., No free lunch theorems for optimization. *Evolutionary Computation*, IEEE Transactions on, 1, 1, 67-82, (1997).
- [8] Cantu-Paz, E., **Efficient and accurate parallel genetic algorithms**. Kluwer Academic Publishers, (2000).
- [9] Martello, S., and Toth, P., **Knapsack Problems**. Wiley, (1990).
- [10] Gupta, J.N.D., and Ho, J.C., A new heuristic algorithm for the one-dimensional bin packing problem. **Production Planning and Control** 10, 6, 598-603, (1999).
- [11] Fleszar, K., and Charalambous, C., Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. **European Journal of Operations Research**, 210, 176–184, (2011).

- [12] Falkenauer, E., A new representation and operators for GAs applied to grouping problems. **Evolutionary Computation**, 2, 2, 123–144, (1994).
- [13] Bozejko, W., Uchroński, M., and Wodecki, M., Parallel hybrid metaheuristics for the flexible job shop problem. **Computers & Industrial Engineering**, 59, 2, 323–333, (2010).
- [14] Falkenauer, E., A hybrid grouping genetic algorithm for bin packing. **Journal of Heuristics**, 2, 1, 5–30, (1996).
- [15] Scholl, A., Klein, R., and Jurgens, C., BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. **Computers and Operations Research**, 24, 7, 627–645, (1997).
- [16] Belov, G., Scheithauer, G., and Mukhacheva, E. A., One-dimensional heuristics adapted for two-dimensional rectangular strip packing. **Journal of the Operational Research Society**, 59, 6, 823–832, (2007).