

Araştırma Makalesi/Research Article

**ALT SINIR TEMELİNE DAYALI AĞIRLIKLI TAVLAMA YÖNTEMİ İLE
KUTULAMA PROBLEMİNİN ÇÖZÜMÜ**
**SOLUTION OF BIN PACKING PROBLEM WITH WEIGHTED ANNEALING METHOD
BASED ON LOWER BOUND**

Neriman İNAK

Bağımsız Araştırmacı, ninak@denizli.bel.tr
<https://orcid.org/0000-0001-9559-8540>

Sezai TOKAT

Prof. Dr., Pamukkale Üniversitesi Bilgisayar Mühendisliği Bölümü, stokat@pau.edu.tr
<https://orcid.org/0000-0003-0193-8220>

Kenan KARAGÜL¹

Dr. Öğretim Üyesi, Pamukkale Üniversitesi Lojistik Bölümü, kkaragul@pau.edu.tr
<https://orcid.org/0000-0001-5397-4464>

Başvuru Tarihi/Application Date: 11.04.2018

Kabul Tarihi/Acceptance Date: 19.07.2018

DOI: 10.30798/makuiibf.414467

Öz

Bu çalışmada bir boyutlu kutulama problemi için yeni bir iki aşamalı, sezgisel ve melez çözüm yöntemi sunulmuştur. Başlangıç çözümü oluşturmak için alt sınıra dayalı sezgisel bir başlangıç çözüm algoritması önerilmiştir. Önerilen sezgisel ile birlikte literatürde yer alan diğer yerleştirme algoritmaları ele alınmış, elde edilen sonuçlar literatürde ulaşılan sonuçlarla karşılaştırılmıştır. Başlangıç çözümü sonrası elde edilen çözüme ağırlıklı tavlama yöntemiyle birlikte yer değiştirme algoritmaları uygulanmış ve kullanılan kutu sayısını minimize etmek amaçlanmıştır. Literatürde yer alan test kümeleri çözülmüş, çözüm süreleri ve elde edilen sonuçlar bilinen en iyi sonuçlarla ve geliştirilen diğer yöntemlerle karşılaştırılmıştır. Literatür ile yapılan karşılaştırmalarda önerilen sezgisel yöntemin daha kısa sürede çözüme ulaştığı gözlemlenmiştir. Ayrıca çözülen test kümesinin 2 örneğinde literatürdeki en iyi bilinen çözümden daha iyi çözümler elde edilmiştir. Bu bağlamda, önerilen iki aşamalı, sezgisel ve melez çözüm yaklaşımının umut vaat eden bir yaklaşım olduğu yapılan analizler ile ortaya koyulmuştur.

Anahtar Kelimeler: Kutulama Problemi, Ağırlıklı Tavlama, Sezgisel Algoritma

Abstract

In this study, a new two-stage, heuristic and hybrid solution method is presented for one dimensional bin packing problem. A heuristic initial solution algorithm based on the lower bound is proposed to create the initial solution. In addition to the proposed heuristics, other placement algorithms in the literature are discussed, and the results obtained are compared with the results obtained in the literature. Swap algorithms together with the weighted annealing method are applied to the results of the initial solutions, and it is aimed to minimize the number of bins used. The test sets in the literature are solved, the resolution times and the results obtained are compared with the best known solution in the literature and other developed methods. In comparisons with the literature, it has been observed that the proposed heuristic method achieves the solutions in a shorter time. In addition, 2 solutions of the solved test set yielded better solutions than the best known solution in the literature. In this context, the analysis suggested that the proposed two-stage, heuristic and hybrid solution approach is a promising approach.

Keywords: Bin Packing Problem, Weighted Annealing, Heuristic Algorithm

¹ Sorumlu yazar

EXTENDED SUMMARY

Background

The bin packing problem, which is a combinatorial optimization problem with some similarities with the classical knapsack problems, is defined in different forms such as stock-cutting, vehicle loading, air cargo loading, container loading, scheduling, knapsack problem depending on the application being discussed. The bin packing problem, which is the problem of how to best locate the objects with the least usable length, area or volume, is classified as one, two or three dimensional according to the number of dimensions used. Regardless of the size of the bin packing problem, the solution is a non-polynomial NP-hard problem. This problem is confronted in many areas such as logistics, textile, and computer science in real life. While there are exact solutions for small scale problems, it is not possible to find an exact solution for large scale problems. Heuristic methods such as genetic algorithms, simulated annealing, weighted simulated annealing, tabu search are the most used solution approaches in the literature to obtain optimal or near optimal solutions for large scale problems.

Research Problem

In this study, one-dimensional bin-packing problem is considered where it is aimed to place objects in a single dimension with a minimum length within a certain length in one dimensional bin packing problems.

Methodology

In this study, a new two-stage, heuristic and hybrid solution method is presented for one dimensional bin packing problem. A heuristic initial solution algorithm based on the lower bound is used to create the initial solution.

As the second stage of the problem solution phase, weights were assigned to the elements together with the weighted annealing method and the initial solutions were rearranged using swapping operations. Along with the weighted annealing and swapping operations, fewer number of bins have been used than the total number of bins used for the initial solution. In this study, Swap (1,0), Swap (1,1) and Swap (1,2) algorithms are used as swapping operations.

Literature Review

For the one dimensional bin-packing problem, a similar two-stage solution method is used by Loh et al (2006) where the initial solutions are obtained by using first fit decreasing, a weighted annealing method is used with swapping operations whereas Swap(2,2) is also used. In our study, the point of origin is to show that better results in a shorter time could be obtained than the results given in Loh et al (2006).

Results and Conclusions

The test sets in the literature are solved, the resolution times and the results obtained are compared with the best known solution in the literature and other developed methods. In comparisons with Loh et al (2006), it has been observed that the proposed heuristic method achieves the solutions in a shorter time. In addition, two solutions of the solved test set yielded better solutions than the best known solution of Loh et al (2006). In this context, the analysis suggested that the proposed two-stage, heuristic and hybrid solution approach is a promising approach. As a further study, this method can be investigated for two-dimensional and three-dimensional bin packing problems. In addition, it may contribute to the development of a new perspective in the research studies on the solutions of combinatorial problems such as traveling salesman problem, knapsack problem, maximum satisfaction problem and vehicle routing problem.

GİRİŞ

Klasik sırt çantası problemleriyle bazı benzerlikleri olan kombinatoriyal bir optimizasyon problemi olan kutulama problemi (BBP: bin packing problem) ele alınan uygulamaya bağlı olarak stok kesme, araç yükleme, hava kargo yükleme, konteynır yükleme, çizelgeleme, sırt çantası problemi gibi farklı biçimlerde tanımlanmaktadır (Sridhar ve diğ.,2017b). Kutuların içerisine en az kullanılmayan uzunluk/alan/hacim kalacak şekilde nesnelerin en iyi nasıl yerleştirileceğinin araştırılması problemi olan kutulama problemi boyut sayısına göre bir boyutlu, iki boyutlu ve üç boyutlu olarak sınıflandırılır (Sridhar ve diğ., 2017a).

Bir boyutlu kutulama problemlerinde (1-D BPP : 1-dimensional bin packing problem); tek boyutta, belirli bir uzunluk içinde en az uzunluk kalacak şekilde nesnelerin yerleştirilmesi ele alınır. Kullanılacak kutulama alanının ve yerleştirilecek nesnelerin iki boyutlu olduğu iki boyutlu kutulama probleminde (2-D BPP: 2-dimensional bin packing problem) bir alan içine en az boşluk kalacak şekilde nesnelerin yerleştirilmesi hedeflenir. En yaygın olarak bilinen kesme ve yükleme problemleri 2-D BPP için örnektir (Ensari, 2007). Üç boyutlu kutulama problemi (3-D BPP : 3-dimensional bin packing problem) çözülmesi oldukça zor kutulama problemidir. Bu problemde kutular ve kutulara yerleştirilecek olan nesneler üç boyutludur. Uygulama da en sık rastlanan 3-D BPP örneği konteynır yükleme problemidir.

Boyutu ne olursa olsun kutulama problemi, çözümünü polinom zamanlı olmayan NP-zor bir problemdir (Garey ve Johnson, 1979). Bu problem gerçek hayatta lojistik, tekstil, bilgisayar bilimleri gibi pek çok alanda karşımıza çıkmaktadır. Küçük ölçekli problemler için kesin çözümler bulunabilirken büyük ölçekli problemler için kesin çözüm bulmak mümkün olmamaktadır. Büyük ölçekli problemler için optimal/optimale yakın çözümleri elde etmek için genetik algoritmalar, tavlama benzetimi, ağırlıklı tavlama benzetimi, tabu araması gibi sezgisel yöntemler literatürde en çok kullanılan çözüm yaklaşımlarıdır.

Kutulama problemlerini çözmek için kutulama problemlerinin boyutları ve problemlerin ortaya çıktığı endüstriyel problemlere bağlı kısıtlarını dikkate alan farklı yaklaşımlar geliştirilmesi çabaları, araştırmacılar için önemli bir araştırma alanını oluşturmaktadır. Bu araştırma sürecinde hem matematiksel hem de sezgisel yöntemlerin geliştirilmesi ve çözüm yaklaşımı olarak kullanma çabaları artarak devam etmektedir.

Kutulama problemine ilişkin teorik ve uygulamalı çalışmalar 1930'lu yıllardan bu yana devam etmektedir. Bu problemin çözümü için ilk matematiksel yöntemler Rusçası 1939 yılında basılan bir çalışmada önerilmiştir (Kantorovich, 1960). Kutulama problemi için Dantzig ve Philip (1960) ve Ford ve Fulkerson (1961) tarafından geliştirilen yöntemlerden yararlanarak Gilmore ve Gomory (1961) sütun oluşturma kavramını geliştirmişlerdir. Coffman, Garey, ve Johnson (1997) tarafından yapılan bir literatür araştırmasında yaklaşım (approximation) algoritmalarının kapsamlı bir incelemesi yapılmıştır. İlk sığan azalan (first fit decreasing) algoritması (ISAA) (Eilon ve Christofides, 1971; Johnson vd., 1974) ve en iyi sığan azalan (best fit decreasing) algoritması (ESAA) (Johnson vd., 1974) başlangıç çözümleri elde etmek için en sık kullanılan iki algoritmadır. Martello ve Toth Prosedürü (MTP), dal-ve-sınır prosedürü bir boyutlu kutulama problemi için kullanılan en eski yöntemlerden biridir (Martello ve Toth, 1987). Bu yöntem pek çok çalışmada karşılaştırma yapmak için en temel referans olarak kullanılmıştır.

Hübscher ve Glover (1994) tarafından eşdeğer birden çok işlemcide birden çok görevin çizelgenmesi probleminde boş zamanların minimizasyonu için bir Tabu Arama algoritması önerilmiştir. Bu çizelgeleme problemi çoklu kutulama probleminin (multiple bin packing problem) özel bir biçimi ile eş biçimlidir. Araştırmacılar etkin bir çeşitlendirme (diversity) yöntemine dayanan genişletilmiş Tabu Arama algoritması ile kutulama problemine oldukça kaliteli çözümler üretmişlerdir.

Falkenauer (1996) kutulama problemi için melez grupta genetik algoritması (HGGA) önermiştir. Scholl ve arkadaşları (1997), çeşitli sınırları, indirgeme prosedürlerini, sezgisel yöntemleri ve yeni bir dallandırma şemasını da bir araya getirerek kesin bir çözüm yöntemi olarak yeni bir dal-sınır prosedürü (BISON) önermişlerdir.

Carvalho (1999) ve Vanderbeck (1999) sütun oluşturma ve dal-sınır yaklaşımlarına dayalı kesin algoritmalar önermişlerdir. Daha sonra Schwerin ve Wäscher (1999), kullanılacak kutu sayısının, stok kesme probleminden türetilmiş yeni bir alt sınırın kullanılması ile MTP'nin çok daha iyi sonuçlar verdiğini göstermişlerdir. Ho ve Gupta (1999) tarafından yapılan çalışmada kutularda kalan toplam ağırlığı en aza indirmek için minimum aylak kapasite (MAK) algoritmasını önermişlerdir. Ho ve Gupta çalışmalarında ISAA, ESAA ve MAK algoritmaları için bir karşılaştırma yapmışlardır. Fekete ve Schepers (2001), kutulama problemi için eşlenik uygunluk fonksiyonlarına (dual feasible functions) dayanan bir alt sınır sınıfı ortaya koymuştur. Böylece, eşlenik uygunluk fonksiyonlarıyla alt sınırlar elde etmek için basit ve hızlı bir yaklaşım önerilmiştir. Fleszar ve Hindi (2002), kutulama problemi için birkaç yeni sezgisel yöntem önermiştir. Önerilen sezgisellerden en etkili olanı değişken komşuluk araması (variable neighborhood search) (DKA) (Hansen ve Mladenović, 1999) meta-sezgiseli ile Fekete ve Schepers (2001) tarafından önerilen yeni alt sınırların kullanılmasına dayanan bir yöntemdir. Önerilen bu sezgiselde sarsımlı minimum aylak kapasite (SMAK) ve DKA bir arada kullanılmaktadır. Elde edilen sonuçlar standart MAK, yeni MAK tabanlı sezgisel, gevşetilmiş (relaxed) MAK (GMAK), SMAK ve örneklemeli (sampling) MAK (ÖMAK) algoritmalarıyla karşılaştırılmıştır. MAK ve sezgisel MAK gibi yaklaşımların en kötü performansı ise Caprara ve Pferschy (2004, 2005) tarafından incelenmiştir.

Alvim, Ribeiro, Vanderbeck ve Alosie (2001) alt sınır tekniklerini de içeren özel bir prosedür olarak melez iyileştirme sezgiselini önermişlerdir. Bu çalışmaya ek olarak yine Alvim ve arkadaşları (2004), yaptıkları bir çalışmada melez iyileştirme sezgiselini eşlenik kutulama problemi için kullanmışlardır. Bhatia ve Basu (2004), çok kromozomlu genetik gösterim yaklaşımı ve daha iyi sığan bir sezgisel yöntem önermişlerdir. Genetik algoritmayı kullanarak kutulama problemini çözmek için çok kromozomlu bir genetik kodlama ve küme tabanlı genetik operatörler önerilen yaklaşımın çekirdeğini oluşturmaktadır. Daha iyi sığan sezgisel yöntemle zenginleştirilmiş genetik algoritmanın performansı HGGA ile karşılaştırılmıştır.

Loh ve arkadaşları (2006) ise ağırlıklı tavlama benzetimini temel alan bir sezgiseli kutulama problemlerinin çözümü için önermişlerdir. Bu çalışmada başlangıç çözümü elde etmek için ISAA kullanılmış, başlangıç çözümlerinin kalitelerinin geliştirilmesi için ağırlıklı tavlama benzetimi ve yer değiştirme operatörleri kullanılmıştır. Bu yöntem melez iyileştirme sezgiseli ile karşılaştırılmıştır (Loh vd., 2006). Singh ve Gupta çalışmalarında genetik algoritma ve sezgisel yaklaşımı birleştiren melez durağan durum gruplamalı genetik algoritma (hybrid steady-state grouping genetic algorithm) H-SGGA + SMAK yöntemi ile diğer MAK algoritmalarını karşılaştırmıştır (Singh ve Gupta, 2007). Kutulama problemlerinde evrimsel algoritmalar Poli (2007) tarafından incelenmiştir. Yapılan çalışmada nesne-boyut histogramıyla kutu-boşluk histogramını eşleştirme kavramına dayanan yeni bir algoritma önerilmiştir. Yaklaşımda, histogramlar arasındaki farkı en aza indirmek için nesnelerin nasıl önceliklendirileceğine karar veren yapısal bir sezgisel mekanizma kullanmışlardır. Böyle bir fonksiyon doğrusal kayıt tabanlı genetik programlama sistemi kullanılarak geliştirilmiştir. Geliştirilen bu yöntem ESAA ile karşılaştırılmıştır. Evrimsel algoritmalarla ilgili Rohlfshagen ve Bullinaria (2007) tarafından da bir çalışma yapılmıştır. Hashim ve arkadaşları tarafından yapılan çalışmada bir boyutlu kutulama probleminin başlangıç çözümlerinin elde edilmesi için kesin (exact), rastgele (random), ilk sığan (first fit) (ISA), en iyi sığan (best fit) (ESA), ISAA ve ESAA yaklaşımları karşılaştırılmıştır (Hashim vd., 2014). Gourgand ve diğ. (2014) tarafından yapılan bir çalışmada HCT (hospital community of territoriality) problemi olarak adlandırılan eylem planlaması ve kaynak ataması sorununu modellemek için

kutulama problemi çözüm yaklaşımı kullanılmıştır. HCT, Fransa'da kurulan bir topluluktur ve farklı yerlerdeki kaynakların organize edilmesi ve planlanması ile kaynak kullanım verimliliğini artırmayı amaçlamaktadır. Burada geliştirilen düşünce, tıp alanındaki sınavların farklı coğrafi bölgelere ve dönemlere atanması ile kaynakların küresel etkinliğinin geliştirilmesini sağlamaktır. Bu problemde eylemlerin mümkün olan en kısa sürede yapılmasını ve bu sayede tahsis edilen toplam sürenin en küçük olmasını sağlamak amacı ile her bir eylem bir kaynağa ve bir periyoda atanır. Bu çalışmada çözüm yöntemi olarak meta-sezgisel ve sığan algoritmalarının (sonraki sığan yer algoritması (next fit algorithm)), ISA, ESA birleşiminden oluşan melez bir yöntem kullanılmıştır (Gourgand vd., 2014). 2016 yılında Delorme, Iori ve Martello (2016) kutulama probleminin tam olarak çözümü için geliştirilen en önemli matematiksel modelleri ve algoritmaları gözden geçirmek ve mevcut en yeni yazılım araçlarının performansını en son teknoloji bilgisayarlarda deneysel olarak değerlendirmek amacıyla bir çalışma yapmıştır. Yine Ozcan ve arkadaşları (2016) yaptıkları bir çalışmada CUDA kullanarak grafik işlem biriminin gücünden yararlanan gruplama genetik algoritmasını önermişlerdir. Sridhar ve arkadaşları (2017b), GA ile çok-kısıtlı kutulama problemleri için etkin çözüm yöntemleri önermişler ve bu önerilen yaklaşımın bu tip lojistik problemlerinin çözümünde etkin bir yaklaşım olduğunu ileri sürmüşlerdir.

Bu çalışmada bir boyutlu kutulama problemi için ağırlıklı tavlama yaklaşımı önerilmiştir. Ağırlıklı tavlama yaklaşımı bir boyutlu kutulama problemi gibi kombinatoriyal bir optimizasyon problemine uygulandığında arama alanının farklı noktalarına sıçramalar oluşturularak arama alanı genişletilir. Bu sıçramalar bir sonraki iterasyonda elde edilen maliyet/kar kazanımlarına bağlı olarak ağırlık atamaları ile kontrol edilir. Ağırlık atama işlemleri yerel arama operatörleri uygulanarak elde edilen çözümler geliştirilir.

1. BAŞLANGIÇ ÇÖZÜMÜ İÇİN KULLANILAN ÇÖZÜM YÖNTEMİ

Kutulama problemleri temel olarak iki aşamalı bir çözüm yaklaşımı kullanılarak çözülür. İlk aşamada uygun bir algoritma kullanılarak başlangıç çözümleri elde edilir. İkinci aşamada elde edilen başlangıç çözümlerine başka bir algoritma uygulanarak optimal/optimale yakın çözümleri elde edilmeye çalışılır. Bu çalışmada başlangıç çözümlerinin elde edilmesi için alt sınır temelli bir sezgisel kullanılacaktır. İkinci aşamada ise bu başlangıç çözümleri önerilen ağırlıklı tavlama benzetimi ve yer değiştirme operatörlerine dayanan bir sezgiselle geliştirilecektir. Bu bağlamda alt sınır kavramı ve önerilen alt sınır temelli başlangıç çözüm yaklaşımı aşağıda verilmiştir. Kutulama problemi için "alt sınır değeri (LB :lower bound)" aşağıdaki gibi açıklanmaktadır.

Kutulama problemi için belirgin bir alt sınır (LB) denklem (1) ile hesaplanabilir.

$$LB = \lfloor \sum_{i \in I} S_i / C \rfloor \quad (1)$$

S_i : i . nesnenin boyutu,

C : Kutu kapasitesi olmak üzere,

LB (Alt Sınır)'nin, boyut S_i 'nin kapasite C 'ye göre yeterince küçük olduğu problemler için "iyi" sonuçlar elde edilmesi beklenebilir. LB aslında, nesnelerin farklı kutular arasında bölünebileceği varsayımına dayanır. Dolayısıyla yaklaşımın getirdiği hata, genellikle verinin süreklilik eğilimi göstermesiyle azalır (Loh vd., 2006).

Alt Sınıra Dayalı Sezgisel Başlangıç Çözüm Algoritması

Başlangıç çözümü oluşturmak için kullanılan alt sınıra dayalı sezgisel başlangıç çözümünde (ASSBC) öncelikle kutulanan nesnelere için denklem (1) kullanılarak bir LB değeri hesaplanır. Hesaplanan LB değeri kadar kutu açılır. Daha sonra nesnelere büyükten küçüğe sıralanır ve birinci nesne

birinci kutuya ikinci nesne ikinci kutuya şeklinde açılan kutulara nesnelere son kutuya kadar yerleştirilir. Son kutuya gelindiğinde sıradaki nesne yine sonuncu kutuya, bir sonraki nesne ise sonuncudan bir önceki kutuya yani $(LB - 1)$ 'nci kutuya yerleştirilir. Algoritma nesne kalmayınca kadar bu şekilde devam eder (Beisiegel vd., 2005). Eğer kutular dolmuş ve elimizde hala yerleştirilemeyen nesne varsa kalan nesnelere için algoritma tekrarlanır. Algoritma 1'de ASSBC'nin sözde kodu verilmiştir.

Algoritma 1: ASSBC Sözde Kodu (Beisiegel ve ark. 2005)

1. Nesnelere kümesi S 'yi büyükten küçüğe sırala.
2. Kullanılacak tüm kutuları kapat.
3. Liste S için en uygun LB hesapla.
4. Nesnelere için LB kadar kutu aç.
5. Listedeki nesnelere sırayla açılmış olan kutulara yerleştir.
6. Kutulara yerleştirilen nesnelere listeden sil.
7. Listede yerleştirilecek nesne kalmadı ise dur.
8. Eğer kapalı kutular kümesi boş değilse adım 3'e git, aksi takdirde kalan kutuları yerleştir.

Bu çalışma kapsamında önerilen başlangıç çözümleri algoritmasının etkinliğini belirlemek üzere üç farklı test problemi kümesi kullanılmıştır. Bu test problemleri OR Library'den alınmıştır (Beasley, 1990). İlk test kümesi Falkenauer (1996) tarafından geliştirilen U serisi olarak adlandırılan, düzgün dağılımlı, 20 adet test problemine sahip, kutu kapasitesinin 150 olarak belirlendiği test kümesi problemleridir. Bu seride U120, U250, U500 ve U1000 olmak üzere 4 farklı örnek kümesi bulunmaktadır ve her bir grup içerdikleri nesne sayısına göre isimlendirilmiştir. Örneğin U120 test kümesi 120 nesneye sahiptir. İkincisi Wäscher ve Gau (1996)'dan alınan 17 probleme sahip Gau test kümesidir. Her problem farklı nesne sayısına sahiptir. Son olarak da J. Schoenfeld'in 28 örnekten oluşan hard28 test kümesi başlangıç sezgisel algoritmasının performans analizi için kullanılmıştır.

Test kümesinde bulunan örnek sayısı N ile ifade edilmiştir. Referans çözümlere ulaşıp ulaşılmadığını gösteren değerler "başarı oranı" sütununda yer almaktadır. Bu sütun "Optimum sonuca ulaşılan örnek sayısı / Toplam örnek sayısı" oranını göstermektedir. Örneğin u120 kümesinin ilk sıranın algoritmasıyla çözümünde hiçbir örnekte optimum sonuca ulaşamamıştır. Bu yüzden başarı oranı $0/20$ 'dir. Yine elde edilen sonuçlara göre mutlak sapma ve bağıl sapma hesaplanmıştır. Mutlak sapma; literatürdeki en iyi bilinen çözüm (best known solution) (EIBC) ile elde edilen çözüm arasındaki farkın mutlak değerinin alınmasıyla elde edilir. Her bir test kümesindeki yine her bir örnek için mutlak sapma denklem (2)'deki gibi hesaplanır. Bağıl sapma denklem (3) ile hesaplanır (Hashim vd., 2014). Bağıl sapma için de ortalama ve maksimum değerler tablolarda gösterilmektedir. Mutlak sapma ve bağıl sapma sırasıyla denklem (2) ve denklem (3) ile hesaplanır.

$$\Delta x = |x_0 - x| \quad (2)$$

$$BS = \frac{\Delta x}{x_0} \quad (3)$$

Denklem (2) ve (3) ait terimlerin açıklanması:

Δx : Mutlak Sapma

BS : Bağıl Sapma

x_0 : En İyi Bilinen Çözüm (EIBC),

x : Bulunan çözüm.

Tablo 1: ASSBC'ye göre sonuçlar

Test Kümesi	N	Başarı Oranı	Mutlak Sapma		Bağıl Sapma		Süre
			Ortalama	Maksimum.	Ortalama.	Maksimum	Ortalama (sn.)
U120	20	0/20	4.55	6	0.090	0.130	0.001
U250	20	0/20	9.40	10	0.092	0.100	0.003
U500	20	0/20	19.60	22	0.097	0.112	0.006
U1000	20	0/20	36.45	42	0.098	0.106	0.011
Gau1	12	5/12	0.58	1	0.040	0.833	0.002
Gau2	5	2/5	0.60	1	0.040	0.909	0.001
Hard28	28	0/28	3.00	5	0.040	0.074	0.002
Ortalama	17.85	1/17.85	10.59	12.42	0.071	0.323	0.004

ASSBC yaklaşımı Visual Studio ortamında C++ programlama dili ile kodlanmıştır. Testler için kullanılan bilgisayar, Windows 7 Enterprise işletim sistemi üzerinde çalışmaktadır ve 4 GB RAM ile 2.26 GHz Intel ® Core™ 2 Duo CPU'ya sahiptir. Tek CPU kullanılmıştır. Tablo 1'de test kümelerine ait ASSBC çözümlerine ait sonuçlar özetlenmiştir.

2. KUTULAMA PROBLEMİ ÇÖZÜMÜNDE AĞIRLIKLI TAVLAMA SEZGİSELİ

Bir boyutlu kutulama probleminde n tane nesne özdeş kutulara yerleştirilmeye çalışılacaktır. Her bir kutunun kapasitesi C 'dir. Amaç, kapasite kısıtlarını ihlal etmeden kutu sayısını en aza indirmektir. Bu amacı sağlamak için ağırlıklı tavlama (weighted annealing) algoritması kullanılmıştır.

2.1. Ağırlıklı Tavlama Kavramı

Ağırlıklı tavlama, kombinasyonel optimizasyon problemlerinin çözümünde kullanılmak üzere Elidan, Ninio ve Schneider (2002) tarafından geliştirilmiştir. İlk olarak 2002 yılında Elidan ve Ninio (2002) genel makine öğrenme problemlerinde geliştirilmiş hipotezleri keşfetmek için küresel aramayı (global search) yerel arama ile birleştiren bir yaklaşım sunmuştur. Bu çalışmada eğitim verilerini bozan yerel maksimumlardan kaçma stratejileri üzerinde durulmuş, eğitim örneklerini yeniden ağırlıklandırmak için basit stratejiler sunulmuştur. Bu bağlamda eğitim örneklerinde W (ağırlık) göz önüne alınacak şekilde puanlar arttırılmıştır (Elidan vd., 2002). Ninio ve Schneider (2005) tarafından yapılan bir çalışmada gezgin satıcı problemi ele alınmış, sistemin her bir bileşenine bir ağırlık atanarak yeni bir ağırlıklı maliyet fonksiyonu tanımlanmıştır. Ağırlıklı tavlama elde edilen sonuçlar açgözlü algoritması (greedy algorithm) ve benzetimli tavlama ile karşılaştırılmış, ağırlıklı tavlamanın benzetimli tavlama göre daha kısa sürede daha iyi sonuçlar verdiği gözlemlenmiştir (Ninio ve Schneider, 2005).

Ağırlıklı tavlama; benzetimli tavlama ve deterministik tavlama gibi diğer meta-sezgisel algoritmalarından farklı bazı özelliklere sahiptir. Özellikle benzetimli tavlama ve deterministik tavlama bir başlangıç çözümüyle başlar ve kötüleşen taşıma içeren yerel aramada bir dizi işlem gerçekleştirir. Bozulan bir taşımaya, amaç fonksiyonu değeri belirli bir eşğin ötesinde kötü olmadığı sürece izin verilebilir. Ağırlıklı tavlama sadece amaç fonksiyonunu dikkate almaz. Aynı zamanda, bir optimizasyon çalışmasının her aşamasında çözülmüş arama alanının ne kadar iyi kullanıldığına bakar. Bir boyutlu kutulama problemi gibi kombinasyonel optimizasyon problemlerine uygulandığında, ağırlıklı tavlama yaklaşımı arama alanının farklı noktalarında bozulmalar oluşturarak komşu aramayı hızlandırmayı ve arama alanını genişletmeyi amaçlar. Bozulmalar bir sonraki iterasyonda elde edilen kazanımlara bağlı olarak ağırlık atamaları ile kontrol edilir. Yerel optimum için arama hızlandırılmış ve çözüm kalitesi ağırlıklı tavlama ile geliştirilmiş olmalıdır (Loh vd., 2006).

2.2. Bir Boyutlu Kutulama Problemi İçin Ağırlıklı Tavlama Algoritması

Bir boyutlu kutulama problemi için ağırlıklı tavlama algoritması Algoritma 2'de verilmiştir.

Algoritma 2: Bir Boyutlu Kutulama Problemi İçin Ağırlıklı Tavlama Algoritması (Loh vd., 2006)

- 1) ISAA algoritması kullanarak başlangıç çözümü oluşturulur.
- 2) Başlangıç çözüme dayanarak kutular için bir dizi ağırlık hesaplaması ve ataması yapılır, her bir kutu için kalan artık kapasiteler hesaplanır.
- 3) Kutuların tüm çiftleri arasında yer değiştirme (swap) işlemi yapılarak yerel arama yapılır. Amaç fonksiyonu, herbir kutuya yerleştirilen nesnelerin toplam büyüklüklerinin karelerinin toplamını maksimum yapmaktır.
- 4) Bir önceki iterasyonun sonuçlarına göre yeni ağırlık dizisi hesaplanarak tekrar ağırlıklandırma yapılır.
- 5) Durma kriterlerine ulaşıncaya kadar 3. adıma geri dönlür.

2.2.1 Amaç Fonksiyonu

Bir boyutlu kutulama problemi için denklem (4)'te ifade edilen amaç fonksiyonu, herbir kutuya yerleştirilen nesnelerin boyutları toplamının karelerinin toplamını maksimize eder (Loh vd., 2006). Amaç fonksiyonunun yapısını gösteren bir örnek Şekil 1'de verilmiştir.

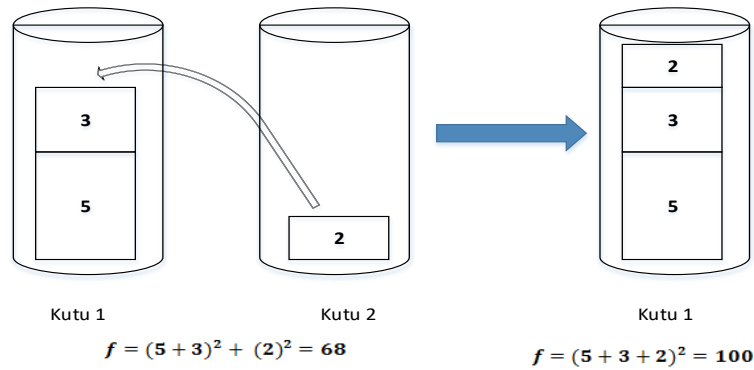
$$Max f = \sum_{i=1}^p (l_i)^2 \quad (4)$$

$$l_i = \sum_{j=1}^{q_i} t_{ij} \quad (5)$$

l_i : i . kutudaki öğelerin boyutlarının toplamını

t_{ij} : i . kutudaki j . öğenin boyutunu

q_i : i . kutudaki öğelerin sayısını temsil etmektedir.



Şekil 1: Kutuların karelerinin toplamını maksimum yapmak

Şekil 1'de 2 farklı kutuda yer alan nesnelerin ağırlıklarına göre amaç fonksiyonu hesaplandığında elde edilen sonuç 68'dir. Ardından yer değiştirme işlemi yapılarak nesneler tek bir kutuda toplanmıştır. Bu durumda amaç fonksiyonu 100 olarak hesaplanmış yani kutuya yerleştirilen nesnelerin boyutlarının kareleri toplamı maksimize edilmiştir.

2.2.2 Ağırlık Atama

Nesnelerin boyutlarındaki değişiklikler herbir kutuya farklı ağırlıklar atanarak elde edilir ve bunlar her iterasyonda tekrar hesaplanır. i kutusuna atanan ağırlık W_i ile gösterilir ve (6) ve (7) denklemleri ile hesaplanır (Loh vd., 2006). Ağırlık hesaplama Şekil 2'deki görsel bir örnek üzerinde gösterilmiştir.

$$W_i = 1 + Kr_i \quad (6)$$

$$r_i = \left(\frac{c - l_i}{c} \right) \quad (7)$$

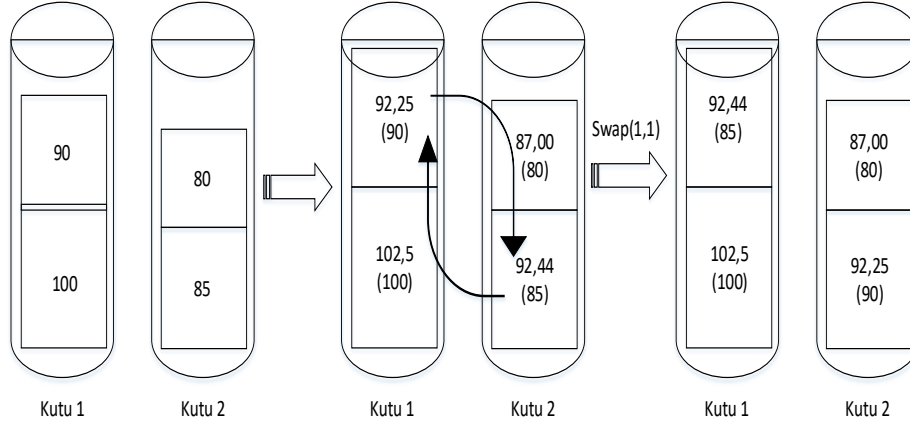
l_i : i . kutudaki nesnelerin boyutları toplamı

r_i : i . kutunun nesnelerin yerleştirilmesinden sonra kalan boş kapasite oranı

C : kapasite

K : sabit değer

$$C = 200 \quad K = 0.5 \quad W_i = 1 + 0.5 \left(\frac{200 - l_i}{200} \right) \quad W_1 = 1.0250 \quad W_2 = 1.0875$$



Şekil 1: Ağırlık hesaplama (weight calculation) (Loh vd., 2006)

$$f' = (102.50 + 92.25)^2 + (87.00 + 92.44)^2 = 70126.3$$

$$f'_{yeni} = (102.50 + 92.44)^2 + (87.00 + 92.25)^2 = 70132.2$$

Şekil 2'de ilk durumda amaç fonksiyonu hesaplanmış daha sonra (6) ve (7) denklemlerine göre bir ağırlık değeri hesaplanarak ve yer değiştirme işlemleri yapılarak kutularda yer alan elemanların mevcut durumu yeniden belirlenmiştir. Bu işlemlerden sonra tekrar amaç fonksiyonu hesaplanmış ve bu değer maksimize edilmeye çalışılmıştır.

2.2.3 Tavlama Benzetimi

Tavlama; malzemeyi belirli bir süre (tavlama sıcaklığına kadar) ısıttıktan sonra, yavaş yavaş soğutmaktır. Tavlama, malzemeyi rahatlatmak, yumuşatmak ve iç yapıyı daha kullanılabilir hale getirmek için yapılan ısı işlemlerinin geneline verilen addır. Isıl işlem, bir katının sıcaklığının belirli bir maksimum dereceye kadar artırılarak tekrar azaltılması işlemidir. Maksimum sıcaklıkta kristalin tüm molekülleri, kendilerini rasgele olarak sıvı faza ayarlar. Sonra, erimiş kristalin sıcaklığı kristal yapı soğutuluncaya kadar düşürülür. Soğuma uygun şekilde yapılırsa kristal yapı çok düzenli olur (Kirkpatrick vd., 1983).

Tavlama benzetimi katıların fiziksel tavlama süreci ile olan benzerlikleri kullanarak geliştirilmiştir. Katıların ısıtılması ve sonra yavaş yavaş soğutulması esasına dayanır. Kirkpatrick ve arkadaşları (1983) tarafından önerilmiştir. Tavlama benzetiminde, amaç fonksiyonunda artışa neden olabilecek komşu hareketler bazen kabul edilerek yerel en iyi noktalarından kurtulmak mümkündür. Amaç fonksiyonunda artışa neden olabilecek bu komşu hareketin kabul edilip edilmemesi, rassal olarak belirlenmektedir. Sıcaklık yüksek olduğunda, amaç fonksiyonunda artışa neden olabilecek hareketlerin kabul edilme olasılığı çok yüksek olacak, sıcaklık düşüldükçe bu olasılık da azalacaktır. Bu sebeple, aramaya yeteri kadar yüksek bir sıcaklık değeri ile başlanmalıdır. Algoritmada, sıcaklık azaltılırken, her sıcaklık değerinde belli sayıda hareket denenerek arama işlemi sürdürülür (Aydın,2013).

Ağırlıklı tavlama yaklaşımında da benzer yöntem kullanılmaktadır. Sürecin başında yüksek sıcaklığa yer verilerek arama alanı genişletilir ve sıcaklık her iterasyonda azaltılır. Ağırlıklı tavlama

farklı olarak bir dizi ağırlık hesaplaması ve ataması kullanılarak yerel arama yapılır.

2.2.4 Ağırlıklı Tavlama Yönteminde Tavlama Süreci

Tavlama süreci için algoritmayı yöneten bir T kontrol parametresi kullanılır. Ağırlık W_i^T olarak belirtilmiştir ve T parametresi algoritmayı şu şekilde kontrol eder (Loh vd., 2006):

- Sürecin başında daha fazla yer değiştirme sağlamak için ağırlık değişikliklerinde yüksek sıcaklığa izin verilir.
- Sıcaklık azalırken her bir nesnenin bozulma miktarı azalır.
- Eğer T 0'a ulaşırsa, algoritma bütün ağırlıkların 1'e eşit olmasıyla süreci sonlandırır.

2.3. Ağırlıklı Tavlama Algoritması İçin Çözüm Kalitesini Arttıran Operatörler

Kutulama probleminde kullanılan yer değiştirme işlemleri ağırlıklı tavlama yaklaşımı ile birlikte uygulanır. Yer değiştirme işlemleri, ağırlıklı tavlama yöntemi ile hesaplanan ağırlık değerleri kullanılarak yer değiştirme fonksiyonları yeniden düzenlenir ve uygulama detayları için okuyucuya ilgili kaynak önerilir (Loh vd.,2006).

2.3.1. Swap (1,0)

$$\Delta f = (l_\alpha * w_\alpha - t_{ai} * w_\alpha)^2 + (l_\beta * w_\beta + t_{ai} * w_\alpha)^2 - (l_\alpha * w_\alpha)^2 - (l_\beta * w_\beta)^2 \quad (8)$$

2.3.2. Swap(1,1)

$$\Delta f = (l_\alpha * w_\alpha - t_{ai} * w_\alpha + t_{\beta j} * w_\beta)^2 + (l_\beta * w_\beta - t_{\beta j} * w_\beta + t_{ai} * w_\alpha)^2 - (l_\alpha * w_\alpha)^2 - (l_\beta * w_\beta)^2 \quad (9)$$

2.3.3. Swap(1,2)

$$\Delta f = (l_\alpha * w_\alpha - t_{ai} * w_\alpha + t_{\beta j} * w_\beta + t_{\beta k} * w_\beta)^2 + (l_\beta * w_\beta - t_{\beta j} * w_\beta - t_{\beta k} * w_\beta + t_{ai} * w_\alpha)^2 - (l_\alpha * w_\alpha)^2 - (l_\beta * w_\beta)^2 \quad (10)$$

2.3.4. Swap(2,2)

$$\Delta f = (l_\alpha * w_\alpha - t_{ai} * w_\alpha - t_{aj} * w_\alpha + t_{\beta k} * w_\beta + t_{\beta l} * w_\beta)^2 + (l_\beta * w_\beta - t_{\beta k} * w_\beta - t_{\beta l} * w_\beta + t_{ai} * w_\alpha + t_{aj} * w_\alpha)^2 - (l_\alpha * w_\alpha)^2 - (l_\beta * w_\beta)^2 \quad (11)$$

2.4. Bir Boyutlu Kutulama Problemi İçin Önerilen Ağırlıklı Tavlama Algoritması Süreci

Bir boyutlu kutulama problemi için önerilen yöntemde öncelikle kullanılacak en az kutu sayısını belirlemek için LB değeri denklem (12) ile belirlenir:

$$LB = \lceil \sum_{i \in I} S_i / C \rceil \quad (12)$$

Başlangıç çözümleri ASSBC algoritması kullanılarak elde edilir. Başlangıç çözümlerini iyileştirmek için ağırlıklı tavlama ile yer değiştirme işlemleri uygulanır. Başlangıç koşulları olarak $T = 1$ ve $K = 0.05$ parametreleri atanır. T sıcaklık değişkeni kullanılarak ağırlık hesabı denklem (13) yardımıyla hesaplanır.

$$w_i^T = (1 + K T_i)^T \quad (13)$$

Başlangıç çözümü belirlendikten sonra sırayla Swap(1,0), Swap(1,1), Swap(1,2) yer değiştirme işlemleri yapılır. Birinci iterasyon bittikten sonra T sıcaklığı $Tred$ oranına bağlı olarak değiştirilir. $Tred = 0.95$ olarak belirlenmiştir. Her iterasyon için T değişkeni denklem (14) yardımı ile tekrar hesaplanır (Loh vd.,2006)

$$T = TxTred \quad (14)$$

Ağırlıklı tavlama yaklaşımıyla önerilen algoritmanın kodu Ek-2'de Algoritma 3 ile verilmiştir.

Tablo 2'de görüldüğü gibi önerilen sezgisel ile Loh'un çalışması arasında farklılıklar vardır. Önerilen sezgiselde başlangıç çözümü olarak ASSBC kullanılırken Loh'un yaptığı çalışmada ilk sığan azalan algoritması (first fit decreasing algorithm) kullanılmıştır. Ayrıca yer değiştirme operatörlerinde de farklılık önerilmiştir. Önerilen yöntemde Swap(2,2) işlemine gerek kalmadan optimum sonuca ulaşıldığı görülmüştür. Bir yer değiştirme operatörünün kullanılmaması ile çözüm süresinin kısaltılması sağlanmıştır. Ancak Loh'un çalışmasındaki yöntem aynı bilgisayar ortamında test edildiğinde Swap(2,2) işlemi olmadan bazı test kümelerinde optimum çözüme ulaşamadığı gözlemlenmiştir. Her iki algoritmadaki testler için kullanılan bilgisayar, 4 GB RAM ile 2.26 GHz Intel ® Core™ 2 Duo CPU'ya sahiptir. Tek CPU kullanılmıştır.

Tablo 2: Önerilen sezgisel ile Loh'un Çalışması arasındaki benzerlikler ve farklar

İşlem	Loh'un Çalışması	Önerilen Sezgisel
Başlangıç Çözümü	ISAA	ASSBC
Uygulanan Algoritma	Ağırlıklı Tavlama	Ağırlıklı Tavlama
Yer Değiştirme İşlemleri	<ul style="list-style-type: none">• Swap(1,0)• Swap(1,1)• Swap(1,2)• Swap(2,2)	<ul style="list-style-type: none">• Swap(1,0)• Swap(1,1)• Swap(1,2)

3. ÖNERİLEN SEZGİSEL YÖNTEMLE ELDE EDİLEN SONUÇLAR

Bu çalışmada U, T, Set, Gau, Was, Hard28 olmak üzere 6 farklı test serisinde 16 farklı test kümesine sahip toplamda 1615 adet problem test edilmiştir. Her test kümesi farklı nesne sayısına, farklı kapasiteye ve farklı zorluk derecesine sahiptir. Kullanılan test kümeleri ve özellikleri Ek-1'de ayrıntılı olarak verilmiştir.

Tablo 3'te U120 test kümesine ait sonuçlar yer almaktadır. Bu tabloda her bir örnek için; kapasite (C), test problemleri için hesaplanan LB değerleri, literatürdeki en iyi bilinen çözüm (EIBC), karşılaştırma için temel alınan Loh'un çalışmasından elde edilen çözümler (WA1BP : Weight Annealing Algorithm for the One-Dimensional Bin Packing Problem), ASSBC ve ağırlık tavlama sezgiseli kullanılarak elde edilen çözümler (Önerilen Sezgisel), bu iki çözüme ait sapma ve bu algoritmaların çözüm süreleri gösterilmektedir.

Tablo 3'te de görüldüğü gibi u120_08 örneğinde sezgisel algoritmalar kullanılarak elde edilen EIBC'de toplamda 51 kutuya yerleştirilirken önerilen sezgisel yöntemle 50 kutuya yerleştirilmiştir. Aynı şekilde u120_19 örneğinde de EIBC sonucu 50 iken önerilen sezgisele göre 49 sonucu elde edilmiştir. Buna göre U120 serisi için EIBC'de toplamda 983 kutu kullanılması gerekirken önerilen algoritmaya göre 981 kutu kullanılabileceği hesaplanmıştır ve bu sonuçlara göre ortalama 0.20 sapma meydana gelmektedir. Böylece önerilen sezgisel sayesinde literatürde yer alan sonuçlardan daha iyi bir sonuç elde edilmiştir. Sürelere bakıldığında ise önerilen sezgisel algoritmanın ortalama 0.08 saniyede WA1BP 'de ise

0.21 saniyede çözüme ulaştığı gözlenmektedir. Buna göre hem toplamda yerleştirilen kutu sayısı hem de çözüm süreleri bakımından önerilen sezgisel algoritmada daha iyi sonuçlar elde edildiği görülmektedir.

Tablo 4'te U, T, Set, Was, Gau ve Hard problem kümelerine ait sonuçlar gösterilmektedir. Tabloda bu problem kümelerinde kaç tane örnek olduğunu gösteren "Örnek Sayısı" sütunu yer almaktadır. Test kümelerindeki örneklerin kaç tanesinde optimum sonuca ulaşıldığını gösteren 2 farklı algoritmanın optimum sonuç sütunları yer almaktadır. Örneğin U120 kümesinde 20 örnek yer almakta ve 2 algoritmada bu 20 örnek için optimum sonuca ulaşılmıştır. Algoritmalar için ortalama çözüm süresi de Tablo 4'te yer almaktadır. WA1BP yönteminde ortalama 0.47 saniyede, önerilen sezgiselde ise ortalama 0.25 saniyede optimum çözümlere ulaşılmıştır. Bu sonuçlara göre optimum çözüm sayıları aynı olmasına rağmen önerilen sezgisel daha hızlı çözüme ulaşılmıştır.

Tablo 3: U120 test kümesi için sonuçlar (Test results for U120)

Test Kümesi	C	LB	EIBC	OPTİMUM SONUÇLAR			ALGORİTMA SÜRELERİ (ms.)	
				WA1BP	Önerilen Sezgisel	Sapma	WA1BP	Önerilen Sezgisel
u120_00	150	48	48	48	48	0.00	5	5
u120_01	150	49	49	49	49	0.00	2	3
u120_02	150	46	46	46	46	0.00	5	4
u120_03	150	49	49	49	49	0.00	10	7
u120_04	150	50	50	50	50	0.00	2	4
u120_05	150	48	48	48	48	0.00	14	5
u120_06	150	48	48	48	48	0.00	30	4
u120_07	150	49	49	49	49	0.00	17	5
u120_08	150	50	51	51	50	1.96	97	17
u120_09	150	46	46	46	46	0.00	89	29
u120_10	150	52	52	52	52	0.00	2	4
u120_11	150	49	49	49	49	0.00	13	6
u120_12	150	48	48	48	48	0.00	16	8
u120_13	150	49	49	49	49	0.00	2	5
u120_14	150	50	50	50	50	0.00	2	3
u120_15	150	48	48	48	48	0.00	8	4
u120_16	150	52	52	52	52	0.00	2	3
u120_17	150	52	52	52	52	0.00	26	11
u120_18	150	49	49	49	49	0.00	9	4
u120_19	150	49	50	50	49	2.00	64	22
Ortalama	150	49.05	49.15	49.15	49.05	0.198	20.75	7.65

Önerilen sezgiseldeki başlangıç çözümünde nesnelere ASSBC'ye göre daha fazla kutuya yerleştirilmiştir ancak yer değiştirme işlemleri bu çözümde daha kısa sürede yapılabilmektedir. Ayrıca önerilen sezgiselde yer değiştirme işlemlerinde Swap(1,0), Swap(1,1), Swap(1,2) kullanılmakta, WA1BP çözümünde Swap(1,0), Swap(1,1), Swap(1,2) ve Swap(2,2) kullanılmaktadır. Önerilen sezgiselde her iterasyon için diğer algoritmaya göre yer değiştirme daha az yapıldığı için çözüm süresi azalmaktadır. Ayrıca önerilen sezgiselde başlangıç çözümünden sonraki yer değiştirme işlemleri başlangıç çözümünün sağladığı faydaya bağlı olarak daha iyi çözüm süreleri sağlamaktadır. WA1BP çözümünde Swap(2,2) işlemini kaldırarak algoritmayı çalıştırdığımızda bazı örneklerde optimum sonuca ulaşılmadığı gözlemlenmiştir.

Tablo 4: U, T, Set, Was, Gau ve Hard test kümeleri için WA1BP ve Önerilen Sezgisel çözümleri

Test Kümesi	Örnek Sayısı	WA1BP		ÖNERİLEN SEZGİSEL	
		Optimum Sonuç	Ortalama süre (sn.)	Optimum Sonuç	Ortalama süre (sn.)
U120	20	20	0.02	20	0.01
U250	20	20	0.16	20	0.09
U500	20	20	0.74	20	0.36
U1000	20	20	2.32	20	2.18
T60	20	20	0.02	20	0.01
T120	20	20	0.07	20	0.03
T249	20	20	0.31	20	0.17
T501	20	20	1.27	20	0.44
SET1	720	720	1.85	720	0.44
SET2	480	480	0.14	480	0.03
SET3	10	10	0.19	10	0.13
WAS1	100	100	0.00	100	0.00
WAS2	100	100	0.00	100	0.00
GAU1	12	12	0.03	12	0.01
GAU2	5	5	0.16	5	0.01
HARD28	28	28	0.21	28	0.10
Toplam	1615	1615		1615	
Ortalama			0.47		0.25

Tablo 5'te ise Loh'un makalesinde yer alan algoritmalar ve çözüm süreleri gösterilmiştir. Tablo 3, 4'te yer alan çözüm süreleri Loh'un WA1BP algoritmasına ait kodların test bilgisayarında yeniden çalıştırılması ile elde edilen sürelerdir. İki algoritma da aynı bilgisayarda çalıştırılıp test edilmiştir.

Tablo 5: Loh'un WA1BP algoritmasına göre çözümü (WA1BP solution of Loh) (Loh vd.,2006)

Test	Örnek	HI_BP		PMBS' + VNS		WA1BP	
		Optimum	Ortalama süre	Optimum	Ortalama süre	Optimum	Ortalama süre
U120	20	20	0.00	20	0.02	20	0.00
U250	20	20	0.12	19	0.03	20	0.03
U500	20	20	0.00	20	0.04	20	0.18
U100	20	20	0.01	20	0.07	20	1.24
T60	20	20	0.37	20	0.01	20	0.00
T120	20	20	0.85	20	0.02	20	0.00
T249	20	20	0.22	20	0.02	20	0.01
T501	20	20	2.49	20	0.06	20	0.03
SET1	720	720	0.19	694	0.15	720	0.17
SET2	480	480	0.01	474	0.10	480	0.19
SET3	10	10	4.60	2	3.74	10	0.13
Toplam	1370	1370		1329		1370	
Ortalama			0.87		0.39		0.18

Tablo 6'da ise Gau'ya ait test kümelerinden elde edilen sonuçlar gösterilmektedir. WA1BP çözümünde 3 örnekte (TEST0058, TEST0068 ve TEST0082) literatürdeki bilinen en iyi sonuçlardan daha iyi sonuçlar elde edilmiştir. Önerilen sezgisel çözümünde ise 1 örnekte (TEST0068) daha iyi sonuç elde edilmiştir. Literatürde 17 örnek için toplamda 290 kutu kullanılmış, WA1BP çözümünde 287, önerilen sezgisel çözümünde de 289 kutu kullanılmıştır.

Tablo 6: Gau test kümesi WA1BP ve Önerilen Sezgisel sonuçları (WA1BP and Suggested Heuristic Results for Gau)

Test Kümesi	EIBC (Kutu Sayısı)	WA1BP (Kutu Sayısı)	Önerilen Sezgisel (Kutu Sayısı)
TEST0005	29	29	29
TEST0014	24	24	24
TEST0022	15	15	15
TEST0030	28	28	28
TEST0058	21	20	21
TEST0065	16	16	16
TEST0068	13	12	12
TEST0082	25	24	25
TEST0044	14	14	14
TEST0049	11	11	11
TEST0054	14	14	14
TEST0055	15	15	15
TEST0055	20	20	20
TEST0075	13	13	13
TEST0084	16	16	16
TEST0095	16	16	16
Toplam		287	289

SONUÇ VE ÖNERİLER

Bu çalışmada bir boyutlu kutulama problemi ele alınmış, ASSBC ile birlikte ağırlıklı tavlama ve yer değiştirme algoritmaları kullanılarak yeni bir çözüm yaklaşımı önerilmiştir.

Kutulama problemi için önerilen bu yöntem iki aşamadan oluşmaktadır. İlk aşamada elemanların kutulara ilk yerleştirilmesini sağlayan başlangıç sezgiseli önerilmiştir.

Problem çözümünün ikinci aşaması olarak, ağırlıklı tavlama yöntemiyle birlikte elemanlara ağırlık ataması yapıp başlangıç çözümleri yer değiştirme işlemleri kullanılarak tekrar düzenlenmiştir. Ağırlıklı tavlama ve yer değiştirme algoritmalarıyla birlikte, başlangıç çözümünde elde edilen toplam kutu sayısından daha az sayıda kutuya yerleştirme yapılmıştır. Bu çalışmada yer değiştirme işlemlerinde Swap (1,0), Swap (1,1) ve Swap (1,2) algoritmaları kullanılmıştır.

Elde edilen sonuçlar Loh'un çalışmasında yer alan WA1BP yöntemiyle karşılaştırılmıştır. Önerilen sezgisel algoritmanın WA1BP yönteminden farkı; başlangıç çözümü olarak farklı bir sezgisel başlangıç algoritması kullanılması ve yer değiştirme işlemlerinde WA1BP yönteminin aksine Swap(2,2) yer değiştirme algoritmasını kullanmadan optimum sonuçlara ulaşabilmesidir. Bu çalışmanın sonucunda iki yöntemin çözüm süreleri karşılaştırılarak önerilen sezgisel yöntemin daha kısa sürede çözüme ulaştığı gözlemlenmiştir.

Tablo 3 incelendiğinde, WA1BP ortalama optimal çözüm sayısı 49.15 ve önerilen sezgisel çözüm için 49.05, sırasıyla ortalama çözüm süreleri incelendiğinde 20.75ms ve 7.65 ms olarak görülmektedir. Bu bağlamda önerilen yöntemin üstünlüğü çözüm kalitesi ve süre olarak ortadadır.

Tablo 4 incelendiğinde bu grup test problemlerinde bulunan optimal çözümler WA1BP ve önerilen sezgisel için eşdeğerdir. Ancak ortalama çözüm süreleri WA1BP için 0.47 sn iken önerilen sezgisel için 0.25 sn olduğu görülmektedir. Bu test kümesi içinde önerilen algoritmanın etkinliği ortadadır.

Tablo 6 incelendiğinde önerilen sezgisel literatürdeki optimal çözümlerin tamamını elde etmiştir. Ancak Loh ve ark. tarafından verilen sonuçlar TEST0058 ve TEST0068 problemleri için literatürdeki çözümlerden daha iyi sonuçlar elde edildiği görülmektedir. Bu bağlamda önerilen yöntemin üstünlüğü çözüm kalitesi ve süre olarak ortadadır. Ayrıca U120 test kümesindeki 2 örnekte literatürdeki EIBC'den daha iyi bir çözüm elde edildiği gözlemlenmiştir.

Önerilen sezgisel yaklaşım sayesinde bir boyutlu kutulama problemi için çözüm süresi kısaltılmış, 2 örnekte de literatürden daha iyi sonuçlar elde edilmiştir. Bu yöntem iki boyutlu ve üç boyutlu kutulama problemleri için araştırılabilir. Ayrıca bu yöntem gezgin satıcı problemi, sırt çantası problemi, maksimum doyum problemi ve araç rotalama problemi gibi kombinatorik problemlerin çözümlerine yönelik arařtırmalarda yeni bir bakış açılarının geliřtirilmesine katkı sağlayabilir.

KAYNAKLAR

- Alvim, A. C. E., Ribeiro, C. C., Glover, F., & Aloise, D. (2001). A hybrid improvement heuristic for the one-dimensional bin packing problem. In *In Extended Abstracts of the 4th Metaheuristics International Conference* (pp. 63–68).
- Alvim, A. C. F., Glover, F., & Aloise, J. J. (2004). A hybrid improvement heuristic for the one-dimensional bin packing problem. *Journal of Heuristics*, 10, 205–229.
- Aydın, İ. (2013). Tavlama Benzetimi Algoritması (Simulated Annealing), Fırat Üniversitesi, *Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı*.
- Beasley, J. (1990). OR-Library. Retrieved April 6, 2016, from <http://people.brunel.ac.uk/~mastjib/jeb/orlib/binpackinfo.html>
- Beisiegel, B., Kallrath, J., Kochetov, Y., & Rudnev, A. (2005). *Simulated annealing based algorithm for the 2D bin packing problem with impurities*. In *Operations Research Proceedings (Hans-Dietrich Haasis, Herbert Köpfer, Jörn Schönberger eds.)*, 309-315, Springer, Berlin, Heidelberg.
- Bhatia A.K. & Basu S.K. (2004) Packing bins using multi-chromosomal genetic representation and better-fit heuristic. In: Neural Information Processing. ICONIP 2004. Lecture Notes in Computer Science, vol 3316, (Pal N.R., Kasabov N., Mudi R.K., Pal S., Parui S.K., eds), 181-186, Springer, Berlin, Heidelberg
- Brandão, F. and Pedroso, J. P. (2016). Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, 69, 56 – 67.
- Caprara, A., & Pferschy, U. (2004). Worst-case analysis of the subset sum algorithm for bin packing. *Operations Research Letters*, 32(2), 159–166.
- Caprara, A., & Pferschy, U. (2005). Modified subset sum heuristics for bin packing. *Information Processing Letters*, 96(1), 18–23.
- Carvalho, J. M. V. (1999). Exact solutions of bin-packing problems using column generation and branch and bound. *Annals of Operations Research*, 86, 629 – 659.
- Coffman, Garey, M. & Johnson, D. (1997). *Approximation algorithms for bin packing problems: A survey*. PWS Publishing, Boston, Massachusetts.
- Dantzig, G. B., & Philip, W. (1960). Decomposition principle for linear programs. *Operations Research*, 8(1), 101–111.
- Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1).
- Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, 17, 259–268.
- Elidan, G., Ninio, M., & Friedman, N. (2002). Data perturbation for escaping local maxima in learning. In *Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*, 132–139, AAAI Press, Edmonton, Alberta, Canada.
- Ensari, M. (2007). *Konteynır yükleme problemine sezgisel bir yaklaşım*. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Bölümü.
- Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2, 5–30.
- Fekete, S. P., & Schepers, J. (2001). New classes of fast lower bounds for bin packing problems. *Mathematical Programming*, 91(1), 11–31.
- Fleszar, K., & Hindi, K. S. (2002). New heuristics for one-dimensional bin-packing. *Computers & Operations Research*, 29(7), 821–839.
- Ford, L. R. J., & Fulkerson, D. R. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6), 849–859.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Company, San Francisco, California.
- Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6), 849–859.
- Gourgand, M., Grangeon, N., & Klement, N. (2014). Activities planning and resource assignment on multi-place hospital system: Exact and approach methods adapted from the bin packing problem. In *Proceedings of the 7th International Conference on Health Informatics*, 117–124, Angers, France.

- Gourgand, M., Grangeon, N., & Klement, N. (2014). An analogy between bin packing problem and permutation problem: A new encoding scheme. In *Proceedings of the IFIP International Conference on Advances in Production Management Systems*, 572–579.
- Hansen, P., & Mladenović, N. (1999). An introduction to variable neighborhood search, In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (S. Voß, S. Martello, I.H. Osman, C. Roucairol eds.), 433-458, Springer, Boston, MA.
- Hashim, N. A., Zulkipli, F., Januri, S. S., & Shariff, S. S. R. (2014). An alternative heuristics for bin packing problem. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 1560-1568.
- Ho, J. C., & Gupta, J. N. D. (1999). A new heuristic algorithm for the one-dimensional bin-packing problem. *Production Planning and Control*, 10(6), 598–603.
- Hübscher, R., & Glover, F. (1994). Applying tabu search with influential diversification to multiprocessor scheduling. *Computers & Operations Research*, 21(8), 877–884.
- Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R., & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithm. *SIAM Journal of Computing*, 3, 299–326.
- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6(4), 366–422.
- Kirkpatrick, S., L., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by simulated annealing, *Science*, New Series, 220(4598), 671-680.
- Loh, K.-H., Golden, B., & Wasil, E. (2006). *Weighted annealing heuristics For solving bin packing and other combinatorial optimization problems: Concepts, algorithms, and computational results*. Doctor of Philosophy Thesis, University of Maryland.
- Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*. 48(2), 256-267.
- Martello, S., & Toth, P. (1987). Algorithms for knapsack problems. *North-Holland Mathematics Studies*, 132, 213–257.
- Ninio, M., & Schneider, J. J. (2005). Weight annealing. *Physica A*, 349, 649–666.
- Oliveira, J. (2003). The Association of European Operational Research Society, Special Interest Group on Cutting and Packing (ESICUP) Datasets. Retrieved April 20, 2016, from <https://paginas.fe.up.pt/~esicup/datasets>
- Ozcan S.O., Dokeroglu T., Cosar A., Yazici A. (2016) A novel grouping genetic algorithm for the one-dimensional bin packing problem on GPU. In *Computer and Information Sciences. ISCIS 2016. Communications in Computer and Information Science*, vol 659. (Czachórski T., Gelenbe E., Grochla K., Lent R. eds), 52-60, Springer, Cham.
- Poli, R., Woodward, J., & Burke, E. K. (2007). A histogram-matching approach to the evolution of bin-packing strategies. In *Proceedings of the IEEE Congress on Evolutionary Computation 2007 (CEC 2007)*, 3500–3507.
- Rohlfshagen, P., & Bullinaria, J. (2007). A genetic algorithm with exon shuffling crossover for hard bin packing problems. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (pp. 1365–1371).
- Scholl, A., Klein, R., & Jürgens, C. (1997). BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, 24(7), 627–645.
- Schwerin, P., & Wäscher, G. (1999). A new lower bound for the bin packing problem and its integration into MTP. *Pesquisa Operacional*, 19, 111–129.
- Singh, A., & Gupta, A. K. (2007). Two heuristics for the one-dimensional bin-packing problem. *OR Spectrum*, 29(4), 765-781.
- Sridhar, R., Chandrasekaran, M., Sriramya, P., Joseph Irudaya Raja, S. (2017a). A review on application of 1D, 2D and 3D bin packing techniques. *Journal of Advanced Research in Dynamical & Control Systems*, 04, 165-169.
- Sridhar, R., Chandrasekaran, M., Sriramya, P., Page, T. (2017b). Optimization of heteogenous bin packing using adaptive genetic algorithm. IOP Conf. Series: Materials Science and Engineering 183, 012026. doi:10.1088/1757-899X/183/1/012026
- Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. *Maths Programming*, 86, 565 – 594.

EKLER

EK 1: Kullanılan test problemleri için detay bilgileri

Adı	Gösterim	Açıklama
Tekdüze (Uniform)	U120,U250,U500, 1000	Kutu kapasitesi $C = 150$ 'dir. Elemanlar tam sayı olup 20 ile 150 arasında tekdüze dağılım göstermektedir. U120 $n = 120$ eleman anlamına gelmektedir. Her $n = 120, 250, 500$ ve 1000 değeri için 20 örneği vardır. Bu problemler Falkneuer tarafından geliştirilmiştir (1996) ve Carvalho tarafından optimal olarak çözülmüştür (1999). Bu problemlere (Beasley, 1990) kaynağından ulaşabilirsiniz.
Üçlü (Triplet)	T60, T120, T249, T501	Kutu kapasitesi $C = 1000$ 'dir. Elemanlar tam sayı olup 250 ile 500 arasında tekdüze dağılım göstermektedir. T60 $n = 60$ eleman anlamına gelmektedir. Her $n = 60, 120, 249$ ve 501 değeri için 20 örneğe sahiptir. Bilinen optimal sonuçlar $n/3$ 'e eşit olduğu için bu problem üçlü olarak adlandırılmıştır. Bu problemler Falkneuer tarafından geliştirilmiştir (1996). Bu problemlere (Beasley, 1990) kaynağından ulaşabilirsiniz.
Set	Set1, Set 2, Set3	Tekdüze dağılım gösteren Set 1 720 örneğe sahiptir. Kutu kapasitesi $C = 100, 120$ ve 150 'dir, $n = 50, 100, 200$ ve 500 'dir. Set 2 480 örneğe sahip olup kutu kapasitesi $C = 1000$ 'dir ve $n = 50, 100, 200, 500$ 'dir. Her kutunun ortalama 3 – 9 elemanı vardır. Set 3 10 örneğe sahip olup kutu kapasitesi $C = 100000$ 'dir. $n = 200$ ve elemanlar 20000 ile 35000 arasında tekdüze dağılım göstermektedir. Set 3 set kümesinin en zoru olarak kabul edilir. Bu problemler Scholl, Klein ve Jürgens tarafından geliştirilmiştir (1997) ve 1184 problemin en iyi duruma getirildiğini bildirmişlerdir. Alvim, Ribeiro, F. Glover, ve D. Aloise (2004) kalan 26 problemler için en iyi sonuçları bildirmişlerdir. Problem kümelerine (Brandão ve Pedrosa, 2016) kaynağında verilen bağlantıdan ulaşabilirsiniz.
Was	Was1, Was2	Was1 100 örneğe sahiptir, kutu kapasitesi $C = 1000$ ve eleman sayısı $n = 100$ 'dür. En küçük eleman boyutu 150, en büyük eleman boyutu 200'dür. Was2 100 örneğe sahip, kutu kapasitesi $C = 1000$ ve eleman sayısı $n = 120$ 'dir. En küçük eleman boyutu 150, en büyük eleman boyutu 200'dür. Bu problemler Schwerin ve Wäscher tarafından geliştirilmiştir(1997, 1999). Bütün problemler optimal olarak çözülmüştür. Bu problem kümelerine (Oliveira, 2003) kaynağından erişebilirsiniz.
Gau	Gau1, Gau2	Gau1 12 örneğe, Gau2 ise 5 örneğe sahiptir. Bu 17 problem Wäscher ve Gau 'dan alınmıştır (1996). Yazarlar tarafından zorlu sorunlar olarak bildirilmektedirler. Bu sorunlardan bazıları optimal olarak çözülmüştür. Problem kümelerine (Oliveira, 2003) kaynağından ulaşabilirsiniz.
Hard28	Hard28	Bu problem kutulama problemi için çok zor 28 örnek içerir. J. Schoenfeld tarafından geliştirilmiştir. Test kümesine [http://www.dcc.fc.up.pt/~fdabrandao/research/arcflow/results/bpp_hard28.html] kaynağından ulaşabilirsiniz.

EK 2: Önerilen algoritma kodu

Algoritma 3: Bir Boyutlu Kutulama Problemi İçin Önerilen Ağırlıklı Tavlama Algoritması

Adım 0. Başlangıç

Parametreler K , $nloop$, T , $Tred$

$K = 0.05$, $nloop = 50$, $T = 1$, $Tred = 0.95$

Girdiler: Nesne sayısı, nesne boyutları, kutu kapasitesi ve alt sınır

Adım 1. ASSBC kullanılarak başlangıç çözümünü oluştur.

Nesneler boyutlarına göre artan olmayan şekilde sıralanır

Do while (nesne listesi boş değilse)

{

Nesnelerin bulunduğu liste \uparrow artmayan şekilde sırala.

Kullanılacak tüm kutuları kapat.

Liste L için en uygun alt sınır (NLB) bul.

Kullanılmak için alt sınır kadar kutu aç.

Listedeki nesnelere öngörülen sırayla açık kutulara yerleştir.

Kutulara yerleştirilen nesnelere listeden kaldır.

Liste boş ise dur.

Eğer kapalı kutular seti boş değilse adım 3'e git, aksi takdirde kalan kutuları yerleştir.

}

Artık kapasiteyi hesapla \uparrow

Adım 2. Mevcut çözümü iyileştir

For $i = 1$ to $nloop$

Ağırlıkları hesapla: $w = (1 - K)^T$

Tüm kutu çiftleri için yap

{

Swap (1,0) uygula

Δf değerlerini değerlendir

if $\Delta f \geq 0$

Nesneyi taşı

Swap(1,0) 'den çık,

Eğer alt sınıra ulaşıldıysa döngüden çık

Swap (1,1) uygula

Δf değerlerini değerlendir

if $\Delta f \geq 0$

Yer değiştir

Swap(1,1) 'den çık,

Eğer alt sınıra ulaşıldıysa döngüden çık

Swap (1,2) uygula

Δf değerlerini değerlendir

if $\Delta f \geq 0$

Yer değiştir

Swap(1,2) 'den çık,

Eğer alt sınıra ulaşıldıysa döngüden çık

}

$T := T \times Tred$

Döngü sonu

Adım 3.

Kullanılan kutu sayısı ve nesnelerin kutulara nihai dağılımı olarak çıktıları ver
