

Sipariş toplama sıklığı düşünceleri altında veri güdümlü depolama yeri atama problemi: Sezgisel bir yaklaşım

Data driven storage location assignment problem considering order picking frequencies: A heuristic approach

İpek ÇOBANOĞLU¹, İrem GÜRE², Vedat BAYRAM^{3*}

¹⁻³Endüstri Mühendisliği Bölümü, Mühendislik Fakültesi, TED Üniversitesi, Ankara, Türkiye.
ipek.cobanoglu@tedu.edu.tr, irem.gure@tedu.edu.tr, vedat.bayram@tedu.edu.tr

Geliş Tarihi/Received: 25.08.2020
Kabul Tarihi/Accepted: 01.03.2021

Düzeltilme Tarihi/Revision: 23.01.2021

doi: 10.5505/pajes.2021.34979
Araştırma Makalesi/Research Article

Öz

Depolama alanları tedarik zinciri yönetiminde kritik öneme sahiptir. Ürün dağıtımını yapmak ve ürünleri depolamak amacıyla kullanılırlar. Bu çalışmada, bir imalat firması tarafından yönetilen bir depolama alanının depolama yeri ataması kararları optimize edilmiştir. Depo yönetim sistemi tarafından kaydedilen tarihsel verileri kullanarak doğrusal olmayan karışık tam sayılı bir problem, yani depolama alanı atama problemini çözmek için bir matematiksel model sunulmuştur. İki ürünün beraber toplanma sıklığı ve her ürünün toplanma sıklığını baz alarak sırasıyla kümeleme ve ABC analizi yapılmıştır ve sonuçlar matematiksel modele yerleştirilmiştir. Aynı zamanda, firmanın depolama yeri problemini çözmek için ağgözlü algoritma geliştirilmiştir. Elde edilen bulgular ışığında, mevcut sistem ve önerilen sistemin G/Ç noktasına olan mesafelerinin karşılaştırılması yapılmış, %49.99'a varan iyileşme görülmüştür.

Anahtar kelimeler: Depolama yeri ataması, Ürün toplama, K-Ortalamalar kümelemesi, ABC analizi, Karışık tam sayılı karesel optimizasyon, Aç gözlü sezgisel.

Abstract

Warehouses are crucial in supply chain management. They are used to distribute and store products. In this study, we optimize storage location assignment decisions in a warehouse managed by a manufacturing firm. A mathematical model is introduced to solve the nonlinear mixed integer optimization problem (NLMIP), i.e., the Storage Location Assignment Problem (SLAP) by using historical data from warehouse management system (WMS). Clustering and ABC analysis are conducted based on the number of times two items are picked together and the picking frequency of items, respectively and results are embedded into our optimization model. Also, a greedy heuristic is developed to solve SLAP of the firm. Analysis results show that there is an improvement of up to 49.99% in total distances between filled slots and the I/O point due to proposed solution compared to that of the current system.

Keywords: Storage location assignment, Order picking, K-Means clustering, ABC analysis, Mixed integer quadratic optimization, Greedy heuristic.

1 Giriş

Depolama alanları, tedarik zincirinin kritik bir parçasıdır ve firmaların verimlilik hedeflerine katkıda buldukları için önemlidirler. Bir depolama alanını verimli bir şekilde yönetmek, ürünlerin mümkün olan en kısa sürede hareket etmesini sağlar ve ürünlerin toplanma süreçlerini geliştirir. Günümüzde firmalar mevcut depolama alanlarını mümkün olduğunca verimli kullanmak için gayret sarfetmektedirler. Depolama alanı yönetimindeki en büyük zorluklardan biri ise ürünlerin çeşitliliğidir. Bu makalenin odağı, bir firmanın tarihsel verilerini kullanarak, çok farklı çeşitlilikteki ürünleri ilgili depolama alanlarına mevcut sistemde olduğundan daha verimli bir şekilde yerleştirmektir. Bunun için ürünlerin depolama alanından en hızlı şekilde toplanmasını sağlayacak kriterler geliştirilmiş, bu kriterleri kullanan bir matematiksel model sunulmuş ve problemi çözmek için bir sezgisel yaklaşım geliştirilmiştir.

1.1 Bir endüstri vaka çalışması

Bu çalışmada bir üretim firmasının depo yerleşim optimizasyonu bir vaka çalışması olarak ele alınmıştır. Firmanın deposu heterojen bir plana sahip olup, 5 farklı bölgeden oluşmaktadır. Deponun heterojen bir plana sahip olmasının nedeni farklı boyutlardaki ürünleri etkin bir şekilde

depolamaktır. Bu çalışmada sadece bölgelerden biri, silo binası esas alınmıştır.

Siloda en büyük ham maddeler, bağlantı parçaları, montaj ürünleri ve ürünlerin bileşenleri depolanmaktadır. Depolanan bazı ürünlere örnek olarak motor, diferansiyel, tekerlek, şanzıman ve plakalar verilebilir. 5 farklı boyutta, toplamda 4,179 adet depolama alanı bulunmaktadır. Analiz edilen 6 aylık süreç boyunca toplamda 805 farklı tipte, 127,189 adet ürün siloda saklanmış, giriş/çıkış yapmıştır. Dolayısıyla, bu çalışma süresince her zaman bu 805 tip ürün kullanılmıştır.

Firmanın kullandığı depolama yöntemi, sınıflandırılmış depolama ve rassal depolama politikalarının karışımıdır. Firma tarafından fiyat temelli bir ABC sınıflandırılması yapılmış ve aktif olarak kullanılmaktadır. Firmanın karışık depolama politikasına sahip olmasının nedeni depolama alanının heterojen olmasıdır. Bu, firma için başlangıçta büyük problemler oluşturmasa da, süreç içerisinde depolama kapasitesi azaldıkça, ürünleri lokasyonlara atama zorluğu ortaya çıkmaktadır. Buna bağlı olarak bazı ürünler rastgele bir şekilde atanmaktadır. Bu durum, belirli ürünler için düzensiz bir yerleşim oluşturmakta ve sonuç olarak ürün toplamada fazladan iş gücü ihtiyacı ortaya çıkmaktadır. Başka bir deyişle, ürünlerin rastgele atanması, aynı siparişte bulunan ürünlerin, yani birlikte sıkça toplanan ürünlerin, toplanma zamanını arttırmaktadır. Ayrıca, planlı bir atama yapılmadığında

*Yazışılan yazar/Corresponding author

depolama alanları verimsiz olarak kullanılmakta ve sonuç olarak ürünler depoya sığmamakta ve bu nedenle başka alanlarda tutulmaya başlanmaktadır.

Depolama alanındaki bazı ürünler belirli bir projeye ait olup bazıları ise genel olarak herhangi bir projede kullanılabilirler. Firmada ürünler toplayıcılar tarafından transfer numaralarına (transferID) göre toplanmaktadır. Bu süreç, bizim çalışmamızda ürünlerin sipariş toplanma sıklığını belirlemek için kullanılacaktır.

Bu çalışmada iki aşamalı bir veri güdümlü optimizasyon metodolojisi ortaya konulmaktadır. Çalışmanın birinci aşamasında ürünlerin çekilme sıklığı ölçütünü baz alan ABC analizi, ürünlerin birlikte çekilme sıklığı ölçütünü baz alan kümeleme analizi, ve ürünlerin proje gruplarının belirlenmesi çalışmaları yer almıştır. İkinci aşamasında ise birinci aşamadan elde edilen girdilerle bir optimizasyon modeli üzerine kurulan sezgisel bir algoritma çalıştırılmaktadır. Süreç akışı Ek B'de görülmektedir.

2 Literatür taraması

Bu çalışmadaki problem literatürde "Depolama Yeri Ataması Problemi" olarak ele alınmıştır. Depolama Yeri Ataması Problemi, ürünlerin depolama alanlarına malzeme taşıma maliyeti ve depolama alanı kullanımının optimize edilerek yerleştirilmesiyle ilgilidir. Depo (ambar) yerleşim planı, depolama kapasitesi, ürün talepleri, depolama politikası, ürünlerin ve depolama alanlarının/raflarının karakteristik özellikleri bu tip problemlerin parametreleri olarak dikkate alınır. Frazelle (1989) Depolama Yeri Ataması Problemini, atanacak ürünlerin sayısı ve depolama alanları özgünlüğünün getirdiği karmaşıklığından dolayı NP-Zor olarak sınıflandırmıştır [1]. Eğer atanacak ürünlerin ve depolama alanlarının sayısı birbirine eşitse, bu problem karesel atama problemi olarak tanımlanır [2]. Atanacak ürünlerin sayısı depolama alanları sayısından fazlaysa ve bir depolama alanına birden fazla ürün yerleştirilebiliyorsa sırt çantası problemi olarak tanımlanmıştır [3].

Kofler ve diğ. (2011) Depolama Yeri Ataması Probleminin amacını, her stok saklama birimini gerekli faktörleri dikkate alarak depolama alanlarına atamak olarak tanımlar [4]. Çalışmada, depolama alanı atamasının iki ana amacı, dolu bir ambara daha fazla stok saklama birimi depolayabilmek ve işlem maliyeti ile iş gücünü azaltmak olarak tanımlanmıştır. Genel olarak yerleştirme sürecinin, stok saklama birimini bir ürün sınıfına atamak ve o sınıfı depolama yerlerine atamak olmak üzere iki aşaması vardır. Makalede rastgele yerleştirme, devir bazlı ölçek ile yerleştirme ve benzerliğe göre yerleştirme olmak üzere üç ana yerleştirme tipinden bahsedilmiştir. Rastgele yerleştirme hariç diğer yerleştirme tipleri bu makalede anlatılan çalışmayla yüzeysel olarak aynı mantıktadır. Devir bazlı ölçek ile yerleştirme, depoda uzun süre durmayan ürünleri kolay ulaşılabilen depolama alanlarına yerleştirmeye dayalıdır. İlk olarak Frazelle ve diğ. (1989) tarafından sunulan benzerliğe göre yerleştirme ise benzer olarak tanımlanan, beraber sıklıkla toplanan ürünleri birbirlerine yakın yerleştirmeye dayalıdır [5]. Frazelle ve diğ. (1989)'nin bu yaklaşımı, bu makalede anlatılacak yerleşim mantığına benzerdir ve çalışmadaki endüstriyel ortağın depolama alanı düzenine uygundur, fakat her depolama alanı düzeni için uygun olmayabilir. Depolama Yeri Ataması Problemi için bir başka matematiksel model ise Muppani ve Adil (2007) tarafından sunulmuştur. Bu modeldeki parametrelerin çoğu envanter tutma masrafı ve ürün toplama masrafı gibi bilgilerden

oluşmaktadır. Problem, Benzetimli Tavlama Algoritması sezgisel yaklaşımı ile çözülmüştür. Bu çalışmada çözüm için sezgisel yaklaşım kullanılmasının nedeni, veri büyüklüğünün çok fazla olmasıdır [6]. Battista ve diğ. (2011), Depolama Yeri Ataması Problemini ele almanın bir diğer yolunu, ürünleri optimize edilmiş depolama alanlarına atama ve ürün taşıma masraflarını en aza indirme amaçlı, depolama yerinin optimizasyonu yapacak yardımcı bir program geliştirilmesi olarak sunmuştur. Bu yöntem, ürünleri giriş/çıkış (G/Ç) bağlantı noktası hareketlerine göre "hızlı hareket eden" (depoda uzun süre kalmayan) ve "yavaş hareket eden" (depoda uzun süre kalan) olarak sınıflandırmaya başlar ve depolama alanlarını G/Ç noktasına olan metrik uzaklıklarına göre "sıcak depolama alanı", "ılık depolama alanı" ve "soğuk depolama alanı" olarak sınıflandırmaya devam eder. Daha sonra, hızlı hareket eden ürünleri sıcak depolama alanlarına atayarak, optimum şekilde ürün-depolama alanı atamasını gerçekleştirir [7].

Kümeleme yöntemi birçok Depolama Yeri Ataması Probleminde kullanılmıştır. Yang ve Nguyen (2015) sınıflandırılmış depolama sistemini ele alarak, ML (must link) ve CL (cannot link) olmak üzere iki kısıt tipli kümeleme analizi sunmuşlardır. Daha sonra, sunulan model COI (cube-per-order index) oranını kullanan bir algoritma ile çözülmüştür [8]. Yang ve Nguyen (2015)'in kullandığı kümeleme yöntemi bu çalışmadakinden farklıdır. Yang ve Nguyen (2015) çalışmalarının gelecekte nasıl geliştirilebileceğine yönelik olarak, bu makalede kullanılmış olan K-ortalama yöntemiyle kümelemeden bahsetmişlerdir [8]. Jane ve Lai (2005) [9], NP-Zor homojen kümeleme modelini gevşeterek geliştirdikleri bir kümeleme yönteminden bahsetmiştir. Bu çalışmada ürünlerin aynı sipariş içinde bulunmasının sıklığını gösteren bir benzerlik ölçüsü sunulmuştur. [9], ürünlerin birlikte toplanma sıklıkları açısından bu çalışmayla benzerlik göstermekle birlikte, [9]'da sunulan çözüm yöntemi bu çalışmada sunulan yöntemden farklıdır. [9]'da sunulan algoritmayı bu makalede sunulan algoritmadan ayıran en büyük fark, yüksek benzerlik ölçüsü olan ürünleri birbirlerinden farklı noktalara koymasındadır. Bindi ve diğ. (2009) özgün bir benzerlik ölçüsü katsayısı sunmuştur. Bu, kümeleme benzerlik katsayısının geliştirilmiş versiyonudur (Jaccard indeksi) [10].

Depolama Yeri Ataması Problemini çözmek için birçok durumda Genetik Algoritma (GA) kullanılır. Büyük ölçekli ve karmaşık problemlerde en kabul edilebilir çözümü bulmak için kullanılan bir yöntemdir. Ene ve Öztürk (2011), bir gruplama ve rotalama problemini çözmek için, gerçek zamanlı operasyonlarda hızlı sonuç verme özelliğinden dolayı bir genetik algoritma geliştirmişlerdir [11]. Norén ve Eriksson, amacı malzeme taşıma zamanını azaltmak olan bir sezgisel yaklaşım sunmuşlardır [12]. Bu yaklaşım en iyi sonuca ulaşmasa da pratikte uygulanabilir olduğundan ve birçok firmada aktif olarak kullanıldığından yine de tercih edilir. Depolama Yeri Ataması Problemlerinde, sezgisel yaklaşım kullanmanın ana sebebi problemin karmaşık olması ve çözümlerle kesin çözüme ulaşılmasının büyük veri setleriyle mümkün olmamasıdır. Dantzig (1957) bu problemin çözümüne yönelik olarak, ağgözlü algoritma kullanan bir yöntem geliştirmiştir [13]. Xu ve diğ. (2008), Depolama Yeri Ataması Problemini deterministik bir yaklaşımla çözmüştür. Bu yaklaşım, ağgözlü komşu ve değişen komşu olmak üzere iki farklı komşu tipinden oluşan bir sezgisel yaklaşım algoritmasıdır [14]. Depolama Yeri Ataması Problemini çözmek

için Wisittipanich ve diğ. (2015) iki farklı sezgi üstü algoritma sunmuştur. Birincisi differansiyel gelişim ve ikincisi ise global yerel ve en yakın komşu parçacık sürü optimizasyonudur [15]. Kim ve diğ. (2020) de depolama yeri ataması problemini sezgisel bir algoritma ile çözmüşlerdir [16].

Bu makalenin literature katkısı şu şekilde sıralanabilir. Öncelikle, depo yönetim sistemi tarafından toplanan tarihsel veri kullanılarak, ürünlerin depodan toplam çekilme miktarına ve birlikte çekilme sıklığına dayalı olan ABC analizi ve K-ortalamlar kümeleme analizi sunulmaktadır. Bu analizlerin sonuçlarını geliştirilen optimizasyon modelinde kullanan, bir veri güdümlü optimizasyon yaklaşımı önerilmektedir. ABC analizi ve K-ortalamlar kümeleme analizlerinin sonuçları ve ürünlerin ait oldukları proje üyelikleri, doğrusal olmayan tam sayılı bir optimizasyon modeline girdi olarak gömülmektedir. Ayrıca, hem aynı proje ve küme üyelikleri olan ürünleri birbirlerine yakın konumlandırarak, hem de bu ürünleri ABC sınıfı üyeliklerine göre, A sınıfı ürünleri G/Ç noktasına yakın olacak şekilde konumlandırarak bir ağırlıklı sezgisel yaklaşım geliştirilmektedir. Geliştirilen yöntemler kullanılarak analizler yapılmaktadır ve firma için depolama alanının kullanımında %43'e varan iyileştirme sağlayan yönetimsel çıkarımlar sunulmaktadır.

Makalenin devamında öncelikle, ABC ve kümeleme analizi uygulamalarından bahsedilmektedir. 4. bölümde depolama yeri ataması problemi tanımlanmaktadır ve oluşturulan NLMIP matematiksel model sunulmaktadır. 5. bölümde, depolama yeri ataması problemini çözmek için geliştirilen ağırlıklı sezgisel yaklaşım algoritması paylaşılmaktadır. 6. bölümde hesapsal sonuçlar ve yönetimsel çıkarımlar paylaşıldıktan sonra son olarak 7. bölümde makalenin sonuçlarından bahsedilmektedir.

3 Betimsel analiz

3.1 ABC analizi

Yongxia (2013) ABC analizini, bir diğer adı Pareto eğrisi olan ABC eğrisinden elde edilen bir yönetim stratejisi olarak değerlendirmiştir. Malzemenin değeri, stok süresi, ürün hacmi ve risk derecesi ABC analizi sınıflandırması için bazı kıstas örnekleridir [17]. Maliyete dayalı bir ABC analizinin bazı dezavantajları vardır. Sık kullanılsa dahi daha yüksek değerli bir ürün A sınıfı bir ürün olarak kabul edilip, G/Ç noktasına yakın bir şekilde yerleştirilir. Oysa sıklıkla toplanan fakat düşük değerli bir ürün C sınıfı olarak düşünülebilir ve G/Ç noktasına daha uzak bir bölgeye yerleştirilebilir. Bu tam olarak bu çalışmanın endüstriyel ortağı ile olan durumdur.

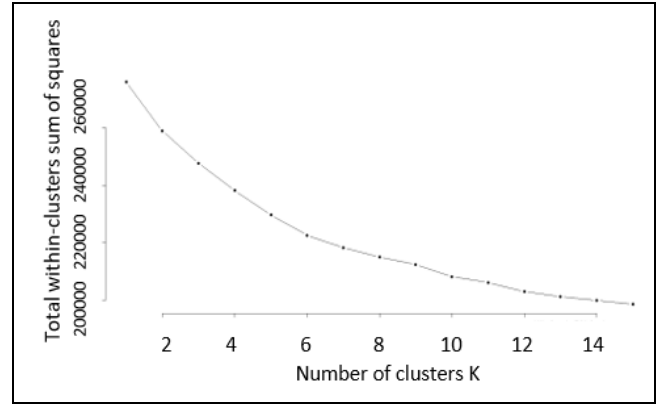
Bu çalışma MS Excel'de VBA kullanılarak yapılan ABC analizi ile başlamaktadır. Temel olarak, en sık toplanılan ürünler sistemde A sınıfı olarak tanımlanmaktadır. Kümülatif sipariş toplama sıklıklarına bakarak, ürünlerin %64'ü A sınıfı, %25'i B sınıfı ve %11'i C sınıfı olarak değerlendirilmiştir. Bunun sonucunda silodaki 805 ürün tipinden 522 tip ürün A sınıfı, 204 tip ürün B sınıfı ve 79 tip ürün C sınıfı olmuştur. Bu şekilde sınıflandırılmış ürünler, optimizasyon modelinde, her sınıfın katsayısını dikkate alarak, A, B ve C sınıfı ürünlerinin bulunduğu depolama alanlarının G/Ç noktasına olan uzaklığını en aza indirmek için kullanılmıştır.

3.2 Kümeleme analizi

Kümeleme analizi için R [18], reshape2 kütüphanesi kullanılmıştır [19]. İlk olarak iki ürünün kaç kez birlikte toplandığı bilgisini içeren bir matris inşa edilmiştir. En sık birlikte toplanan ürünlerin aynı kümede olması

beklenmektedir. R'daki kümeleme analizi için bu matrisi elde etmek üzere Python'da bir kod yazılmıştır. Firma, transferID'ler aracılığıyla sipariş toplama bilgilerinin bulunmasını sağlamıştır. Yazılan Python kodu, temelde ürünlerin transferID'lerini gruplayıp ürünlerin kaç kez bir arada toplandığını saymaktadır.

Daha sonra, öklid mesafeleri kullanılarak R'da kümeleme analizi yapılmıştır. Silodaki 805 tip ürün için maksimum küme sayısı 15 olarak belirlenmiştir. Optimum küme sayısına karar vermek için çizilen dirsek (elbow) yöntemi grafiği Şekil 1'de görülebilir. Bu grafiğe göre, optimum küme sayısı 10 olarak belirlenmiştir çünkü 10 kümeden sonra eğrinin düzleştiği, yani marjinal faydanın azaldığı görülebilir.



Şekil 1. Elbow yöntemi grafiği.

Figure 1. Elbow method graph.

Bu kümeler optimizasyon modelinde aynı kümeye üye olan her ürünün depolama yerleri arasındaki mesafeyi en aza indirmek için kullanılmıştır.

4 Problem tanımı ve geliştirilen model

Depolama Yeri Ataması Problemi, sipariş toplama sürelerini azaltacak şekilde hangi depolama alanlarının kullanılacağına ve hangi ürünlerin hangi depolama alanlarına yerleştirileceğine karar verir. Bizim problemimizin ana amaçları kullanılan depolama alanı sayısını, aynı projeye ait olan ürünlerin arasındaki mesafeyi, birlikte sıkça çekilen ürünlerin arasındaki mesafeyi ve A, B, C sınıfı ürünlerin bulunduğu depolama alanlarının G/Ç noktasına olan mesafelerini azaltmaktır.

Problemi tanımlamaya yarayan setler, parametreler ve değişkenler Tablo 1'de görülebilir.

Depolama alanları kümesi S, ürün tipleri kümesi I, projeler kümesi P, ürün kümeleri C ve A, B, C sınıfı ürünlerin kümeleri sırasıyla A, B, C olarak tanımlanmıştır. İndeksler $(s, s') \in S$, $(i, j) \in I$, $k \in K$ ve $p \in P$ olarak tanımlanmıştır. Ürün tipi i 'nin $(i \in I)$ hacmi h_i , depolama alanı s 'nin $(s \in S)$ hacmi H_s , depolama alanı $s \in S$ ve G/Ç noktası arasındaki mesafe d_{so} , depolama alanı $s \in S$ ve depolama alanı $s' \in S$ arasındaki mesafe $d_{ss'}$ ve son olarak depolanacak ürün tipi i 'nin toplam miktarı w_i olarak tanımlanmıştır. Amaçları öncelik sırasına koymak için amaç fonksiyonunda skalar çarpanlar kullanılmıştır. A sınıfının çarpanı α , B sınıfının çarpanı β , C sınıfının çarpanı δ , kümelerin çarpanı λ , projelerin çarpanı μ ve kullanılan depolama alanlarının çarpanı γ olarak tanımlanmıştır. $\mu > \lambda > \alpha > \beta > \delta > \gamma$ şeklinde sıralanmıştır.

Tablo 1. Problem notasyonu.
Table 1. Notation of the problem.

Setler	
S	Depolama alanları
I	Ürün tipleri
P	Projeler
K	Kümeler
A	A sınıfı ürün tipleri
B	B sınıfı ürün tipleri
C	C sınıfı ürün tipleri
İndeksler	
s, s'	$\in S$
i, j	$\in I$
k	$\in K$
p	$\in P$
Parametreler	
h_i	Ürün tipi i 'nin ($i \in I$) hacmi
H_s	Depolama alanı s 'nin ($s \in S$) hacmi
d_{so}	Depolama alanı $s \in S$ ve G/Ç noktası arasındaki mesafe
$d_{ss'}$	Depolama alanı $s \in S$ ve depolama alanı $s' \in S$ arasındaki mesafe
w_i	Ürün tipi i 'nin toplam miktarı
Skalar Çarpanlar	
α	A sınıfı çarpanı
β	B sınıfı çarpanı
δ	C sınıfı çarpanı
λ	Kümelerin çarpanı
μ	Projelerin çarpanı
γ	Kullanılan depolama alanlarının çarpanı
Değişkenler	
y_s	1, eğer depolama alanı $s \in S$ kullanılmışsa 0, kullanılmamışsa
x_{is}	1, eğer ürün $i \in I$ depolama alanı s 'ye ($s \in S$) atanmışsa 0, atanmamışsa
z_{is}	Ürün tipi i 'nin ($i \in I$) depolama alanı s 'ye ($s \in S$) atanan miktarı

Bu çarpanların ağırlığı firmanın tercihlerine göre değiştirilebilir. y_s , x_{is} ve z_{is} olmak üzere 3 karar değişkeni tanımlanmıştır. y_s , eğer depolama alanı $s \in S$ kullanılmışsa 1, kullanılmamışsa 0 değerini alan bir ikili değişkendir. x_{is} , eğer ürün $i \in I$ depolama alanı s 'ye ($s \in S$) atanmışsa 1, atanmamışsa 0 değerini alan bir ikili değişkendir. z_{is} , ürün tipi i 'nin ($i \in I$) depolama alanı s 'ye ($s \in S$) hangi miktarda atandığını gösteren bir tam sayı değişkenidir.

Geliştirilen model aşağıdadır:

$$\min \mu \sum_{m=1}^m \sum_s \sum_{s'} \sum_{i \in P_p} \sum_{j \in P_p} d_{ss'} z_{is} x_{is} x_{js'} + \lambda \sum_{n=1}^n \sum_s \sum_{s'} \sum_{i \in K_k} \sum_{j \in K_k} d_{ss'} z_{is} x_{is} x_{js'} \quad (1)$$

$$+ \alpha \sum_{i \in A} \sum_s d_{so} z_{is} x_{is} + \beta \sum_{i \in B} \sum_s d_{so} z_{is} x_{is} + \delta \sum_{i \in C} \sum_s d_{so} z_{is} x_{is} + \gamma \sum_{s \in S} y_s$$

k.s.

$$\sum_{s \in S} x_{is} \geq 1 \quad \forall i \in I \quad (2)$$

$$\sum_{i \in I} h_i z_{is} \leq H_s * y_s \quad \forall s \in S \quad (3)$$

$$z_{is} \leq w_i x_{is} \quad \forall i \in I, s \in S \quad (4)$$

$$\sum_{i \in I} x_{is} \leq y_s \quad \forall s \in S \quad (5)$$

$$\sum_{s \in S} z_{is} = w_i \quad \forall i \in I \quad (6)$$

$$x_{is} \in \{0,1\} \quad \forall s \in S, i \in I \quad (7)$$

$$y_s \in \{0,1\} \quad \forall s \in S \quad (8)$$

$$z_{is} \text{ tam sayı} \quad \forall i \in I, s \in S \quad (9)$$

Amaç fonksiyonunun (1) ilk terimi projelerle alakalıdır ve aynı projeye ait ürünlerin birbirlerine yakın depolama alanlarına atanmasını sağlar. Amaç fonksiyonunun bu bölümü, firmanın önem sırasına göre bu çalışmanın en önemli bölümüdür ve en yüksek katsayılı, μ , almıştır. Amaç fonksiyonunun ikinci terimi, kümelerle alakalıdır. Kümeleme analizine göre aynı kümede olan yani birlikte çok sık çekilen ürünlerin birbirlerine yakın konumlandırılmasını sağlar. İkinci en büyük katsayı, λ , bu fonksiyona aittir. Amaç fonksiyonunun üçüncü, dördüncü ve beşinci terimleri sınıf fonksiyonudur. A, B ve C sınıfı ürünlerin G/Ç noktasına olan mesafelerini azaltmayı sağlar. Sırasıyla A, B, C sınıfı katsayıları $\alpha > \beta > \delta$ olarak sıralanmıştır. Bu sayede A sınıfı ürünleri G/Ç noktasına en yakın lokasyonlara, B sınıfı ürünleri A sınıfı ürünlerinden biraz daha uzağa ve C sınıfı ürünleri G/Ç noktasına diğerlerinden de uzak olacak şekilde atamayı sağlar. Bu fonksiyon projeye ve küme fonksiyonlarından sonraki en önemli fonksiyondur. Bu yüzden de katsayıları, projeye ve küme fonksiyonu katsayılarından, μ ve λ , küçüktür. Amaç fonksiyonunun son terimi olan beşinci terim, kullanılan depolama alanı sayısı fonksiyonudur ve kullanılan depolama alanı sayısını azaltmayı sağlar. Bu çalışmada öncelik sıralamasında en sonda olduğu için katsayısı, γ , en düşük katsayı değerini almıştır.

(2), her ürün tipinin en az bir depolama alanına atanmasını sağlar. İkinci kısıt (3), bir depolama alanına atanan toplam ürün hacminin, depolama alanının hacmini geçmemesini sağlar. Üçüncü kısıt (4) ürünlerin sadece kullanılmasına karar verilen depolama alanlarına atanmasını sağlar ve atanan ürün miktarının, o ürün tipinin toplam miktarını geçmemesini sağlar. Dördüncü kısıt (5) her depolama alanına en fazla bir ürün tipinin atanmasını garantiler. (6) her ürün tipinin tüm miktarının yerleştirilmesini garantiler. (7), (8) ve (9) değişkenlerin tanım kümelerini belirtir.

Modelin amaç fonksiyonu doğrusal olmayan bir fonksiyondur. Amaç fonksiyonu için iki aşamalı bir doğrusallaştırma uygulanmaktadır. Birinci aşamada atama değişkenlerinin çarpımından kaynaklanan doğrusal olmayan terim doğrusal hale getirilmektedir. Bunun için $x_{is}, x_{js'}$ değişkenlerinin yerine geçecek olan $b_{isjs'}$ yardımcı değişkeni tanımlanmakta ve ilave kısıtlar eklenmektedir. Modelin bu hali aşağıda sunulmuştur:

$$\begin{aligned} \min \mu & \sum_{m=1}^m \sum_s \sum_{s'} \sum_{i \in P_p} \sum_{j \in P_p} d_{ss'} z_{is} b_{isjs'} \\ & + \lambda \sum_{n=1}^n \sum_s \sum_{s'} \sum_{i \in K_k} \sum_{j \in K_k} d_{ss'} z_{is} b_{isjs'} \\ & + \alpha \sum_{i \in A} \sum_s d_{so} z_{is} x_{is} + \beta \sum_{i \in B} \sum_s d_{so} z_{is} x_{is} \\ & + \delta \sum_{i \in C} \sum_s d_{so} z_{is} x_{is} + \gamma \sum_{s \in S} y_s \end{aligned} \quad (10)$$

k.s.

$$b_{isjs'} \leq x_{is} \quad \forall i, j \in I, \quad s, s' \in S \quad (11)$$

$$b_{isjs'} \leq x_{js'} \quad \forall i, j \in I, \quad s, s' \in S \quad (12)$$

$$b_{isjs'} \geq x_{is} + x_{js'} - 1 \quad \forall i, j \in I, \quad s, s' \in S \quad (13)$$

$$b_{isjs'} \geq 0 \quad \forall i, j \in I, \quad s, s' \in S \quad (14)$$

(2)-(9)

Modelin bu halinde amaç fonksiyonunda tamsayı bir değişken olan z_{is} ile sürekli bir değişken olan ve ancak 0 ya da 1 değerini alan $b_{isjs'}$ değişkeninin çarpımından kaynaklanan ve tamsayı bir değişken olan z_{is} ile sürekli bir değişken olan ve ancak 0 ya da 1 değerini alan x_{is} değişkeninin çarpımından kaynaklanan doğrusal olmayan terimler vardır. Bu terimleri doğrusal hale getirmek için $\bar{z}_{isjs'} \geq 0$ ve $\hat{z}_{is} \geq 0$ yardımcı değişkenleri kullanılmaktadır. Modelin doğrusal hale getirilmiş son hali aşağıda sunulmuştur:

$$\begin{aligned} \min \mu & \sum_{m=1}^m \sum_s \sum_{s'} \sum_{i \in P_p} \sum_{j \in P_p} d_{ss'} \bar{z}_{isjs'} \\ & + \lambda \sum_{n=1}^n \sum_s \sum_{s'} \sum_{i \in K_k} \sum_{j \in K_k} d_{ss'} \bar{z}_{isjs'} \\ & + \alpha \sum_{i \in A} \sum_s d_{so} \hat{z}_{is} + \beta \sum_{i \in B} \sum_s d_{so} \hat{z}_{is} \\ & + \delta \sum_{i \in C} \sum_s d_{so} \hat{z}_{is} + \gamma \sum_{s \in S} y_s \end{aligned} \quad (15)$$

k.s.

$$\bar{z}_{isjs'} \leq w_i b_{isjs'} \quad \forall i, j \in I, \quad s, s' \in S \quad (16)$$

$$\bar{z}_{isjs'} \geq z_{is} - (1 - b_{isjs'}) w_i \quad \forall i, j \in I, \quad s, s' \in S \quad (17)$$

$$\bar{z}_{isjs'} \leq z_{is} \quad \forall i, j \in I, \quad s, s' \in S \quad (18)$$

$$\bar{z}_{isjs'} \geq 0 \quad \forall i, j \in I, \quad s, s' \in S \quad (19)$$

$$\hat{z}_{is} \leq w_i x_{is} \quad \forall i \in I, s \in S \quad (20)$$

$$\hat{z}_{is} \geq z_{is} - (1 - x_{is}) w_i \quad \forall i \in I, s \in S \quad (21)$$

$$\hat{z}_{is} \leq z_{is} \quad \forall i \in I, s \in S \quad (22)$$

$$\hat{z}_{is} \geq 0 \quad \forall i \in I, s \in S \quad (23)$$

$$\bar{z}_{isjs'} \geq 0 \quad \forall i, j \in I, \quad s, s' \in S \quad (24)$$

(2)-(9), (11)-(14)

5 Çözüm yöntemi

Zorluk açısından, Frazelle (1989) Depolama Yeri Ataması Problemini NP-Zor olarak sınıflandırmaktadır [1]. Bu çalışmada, ürün tiplerinin sayısı depolama alanlarından daha azdır. Ayrıca, her depolama alanı yalnızca bir tip ürün depolayabilir. Bu problemin orijinal formülasyonu doğrusal olmayan bir amaç fonksiyonuna, 6,732,369 değişkene (3,368,274 ikili değişken ve 3,364,095 tamsayı değişkeni) ve 10,106,432 kısıta, problemin doğrusal hale getirilmiş versiyonu ise 1.018×10^{14} kısıta, 2.263×10^{13} değişkene sahiptir. Gerçek verilerle kesin çözüm elde etmek için öncelikle ticari çözücü olan CPLEX kullanılması düşünülmüştür. Fakat problemdeki veri büyüklüğünün yol açtığı değişken ve kısıt sayısının yüksek olması nedeniyle, optimal çözümü elde etmek bu şekilde mümkün değildir. Çünkü bu durumda ya çözüm süresi çok uzun olmakta ya da bazı durumlarda, bellek sorunları yüzünden problem çözülememektedir. Çalışmadaki gerçek verinin boyutu 64,924 KB olup problem, gerçek verilerle 1.60 GHz CPU ve 8 GB RAM'i olan bir bilgisayarda Windows işletim sistemi altında CPLEX çözücüsü kullanılarak GAMS üzerinden çözülmeye çalışılmıştır. Çözücüler 10 saat sonunda henüz modeli oluşturmayı tamamlamamışlar ya da bellek sorunu nedeniyle durmuşlardır. Modeli çözmeye başlayamadığı için gerçek veri boyutu 794 KB'a indirilip küçültülerek bir daha denenmiştir. Bu durumda çözücüler, 114 dk.'nın sonunda bellek sorunu yaşayıp çözmeyi durdurmuştur. Bu nedenle, problemi çözmek için, Java yazılımı kullanılarak ağgözlü sezgisel yaklaşım yöntemi geliştirilmiştir. Guerriero ve diğ. (2015)'te depolama yeri ataması problemini çözmek için sezgisel bir yöntem ile CPLEX'i karşılaştırmışlardır ve hesaplama süresi bakımından sezgisel yöntemin daha avantajlı olduğunu saptamışlardır [20].

5.1 Ağgözlü sezgisel yaklaşım

Açgözlü Algoritma, faydayı yüksek tutarak en iyi çözüme ulaşmak için adım adım olurlu bir çözüm oluşturacak şekilde yapılandırılmıştır. En yüksek katsayıyı, kazancı en çok sağlayacak olana atamaya odaklıdır. Açgözlü algoritma, kolay kullanımından, düşük hesaplama maliyetlerinden ve çok sayıda probleme uyarlanabilir olmasından dolayı oldukça sık kullanılır [21].

Firma için en yüksek öncelik aynı projedeki ürünlerin birbirine yakın yerleştirilmesi olduğundan, önerilen açgözlü algoritma, ürünleri öncelikle proje üyeliklerine göre yerleştirir. Proje üyeliği olan ürünlerin tamamı yerleştirildikten sonra, proje üyeliği olmayan ürünler küme üyeliklerine göre yerleştirilir. Projeler ve kümelerin hangi sıraya göre yerleştirileceği ise, sahip oldukları A grubu ürün sayısına göre belirlenir. A grubu sayısı en fazla olan proje/küme, en öncelikli proje/kümedir. Algoritmada her boş depolama yeri için bir metrik hesaplanır. Bu da boş depolama yerinin, sıradaki yerleştirilecek ürünle aynı projeye ait bir başka ürünün yerleşmiş olduğu en yakın dolu depolama yerine olan uzaklığı ve aynı boş depolama yerinin G/Ç noktasına olan uzaklığının toplamıdır. Ürünlerin atanması, her proje ve her kümede bulunan A sınıfı ürünleri sayma ile başlar. İkinci olarak, projeleri ve kümeleri, içerdiği A sınıfı ürün sayısına göre azalan bir düzende sıralar. Daha sonra, sıralama sonucuna göre ilk projeden başlayarak, o proje ürününü G/Ç noktasına en yakın yere atar. Bir sonraki ürünün atanması için sezgisel yöntem, en küçük metriğe sahip depolama yerini seçer. Sonraki ürün bir projeye ait değilse, sıralamadan sonraki ilk kümeden başlayarak, belirli kümenin ürününü en küçük metriğe sahip olan depolama yerine atar. Atamalar yapılırken depolama alanlarının kapasiteleri ve sadece bir tip ürün barındırıyor olması dikkate alınır. Sezgisel algoritmanın sözde kodu Ek A'da görülebilir.

Bu adımlar izlenerek, aynı projeye ait ürünler birbirine yakın tutulmaktadır. Belirli bir projeye ait olmayan genel ürünler için, küme grupları göz önünde bulundurularak onlar da birbirlerine yakın tutulmaktadır. Algoritma, herhangi bir aşamada, A sınıfı ürünleri mümkün olduğunca G/Ç noktasına yakın tutmaya çalışmaktadır. Bunları yaparken, modelin tüm kısıtları sağlanır. Algoritma, tüm depolama yerleri doldurulana kadar veya ürünlerin tümü atanana kadar çalışır.

5.2 Model geçerliliği ve doğruluğu

Öncelikle model doğrulanıp kodun doğru çalıştığı teyit edilmiştir. Bunun için 9 KB boyutunda küçük bir örnek oluşturulmuş ve küçük örnek 2.50 GHz CPU ve 8 GB RAM'i olan bir bilgisayarda Windows işletim sistemi altında CPLEX çözücüsü kullanılarak çözülmüştür ve 68.05 sn. sonunda optimum sonuç elde edilmiştir. Sonuçları karşılaştırmak için küçük örnek, geliştirilen sezgisel algoritma ile bir daha çözülmüştür. Sezgisel algoritma, 0.024 sn. sonunda çözüme ulaşmıştır. Daha sonra, hem kesin çözüm sonucu hem de sezgisel algoritmadan elde edilen çözüm sonuçları karşılaştırılmıştır. Küçük örnek 7 ürün, 10 depolama alanı, 3 proje ve 3 kümeden oluşmaktadır. Her depolama alanının hacmi aynı ve 2,000 br³ kabul edilmiştir. Ürün sınıflandırması Tablo 2'de, depolama alanlarının G/Ç noktasına olan mesafeleri Tablo 3'te ve depolama alanlarının birbirleri arasındaki mesafeler Tablo 4'te görülmektedir.

Tablo 2. Ürün sınıflandırması.

Table 2. Item classification.

Ürün Tipi Numarası	Proje	Küme	ABC Sınıfı	Toplam Miktar	Ürün Hacmi (br ³)
1	1	1	A	2	25
2	2	1	A	1	120
3	3	1	A	5	16
4	3	3	A	10	18
5	1	3	C	1	70
6	2	2	B	3	11
7	2	2	B	6	75

Tablo 3. Depolama alanlarının G/Ç noktasına olan mesafeleri.

Table 3. Distances between I/O point and the slots.

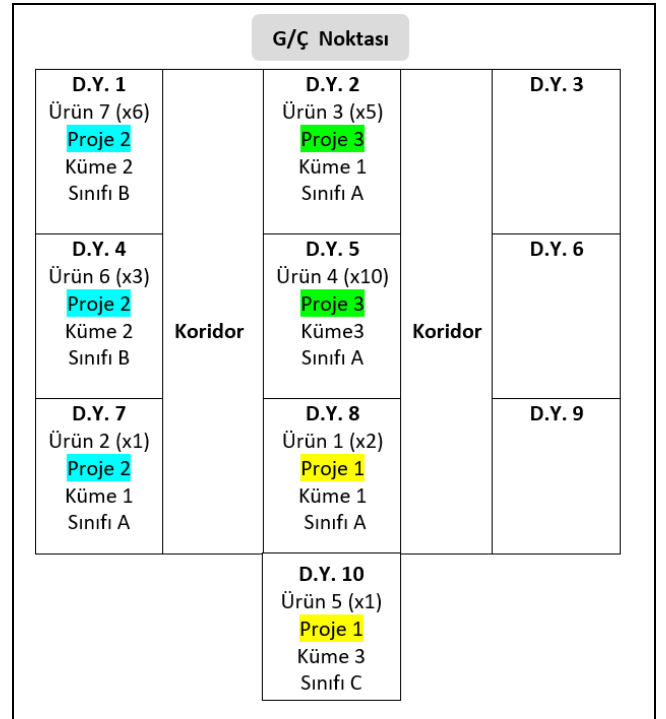
Depolama Alanı (D.A.)	G/Ç Noktasına Olan Mesafe (br)
D.A. 1	2
D.A. 2	1
D.A. 3	2
D.A. 4	4
D.A. 5	3
D.A. 6	4
D.A. 7	5
D.A. 8	4
D.A. 9	5
D.A. 10	5

Tablo 4. Depolama alanlarının birbirleri arasındaki mesafeler.

Table 4. Distances between each slot.

	D.A. 1	D.A. 2	D.A. 3	D.A. 4	D.A. 5	D.A. 6	D.A. 7	D.A. 8	D.A. 9	D.A. 10
D.A. 1	0	5	10	1	6	11	2	7	12	8
D.A. 2	5	0	5	6	1	6	7	2	7	3
D.A. 3	10	5	0	11	6	1	12	7	2	8
D.A. 4	1	6	11	0	5	10	1	6	11	7
D.A. 5	6	1	6	5	0	5	6	1	6	2
D.A. 6	11	6	1	10	5	0	11	6	1	7
D.A. 7	2	7	12	1	6	11	0	5	10	6
D.A. 8	7	2	7	6	1	6	5	0	5	1
D.A. 9	12	7	2	11	6	1	10	5	0	6
D.A. 10	8	3	8	7	2	7	6	1	6	0

Elde edilen kesin ve sezgisel çözümler istenilen kriterler açısından incelenmiştir. Şekil 2 sezgisel çözümü, Şekil 3 ise kesin çözümü göstermektedir. Depolama yerlerine atanan ürün sayısı ve diğer ürün özellikleri (proje, küme ve ABC sınıfı) şekillerdeki hücrelerde görülmektedir.



Şekil 2. Sezgisel çözüm.

Figure 2. Heuristic solution.

G/Ç Noktası				
D.Y. 1 Ürün 6 (x3) Proje 2 Küme 2 Sınıfı B	Koridor	D.Y. 2 Ürün 4 (x10) Proje 3 Küme3 Sınıfı A	Koridor	D.Y. 3
D.Y. 4 Ürün 7 (x6) Proje 2 Küme 2 Sınıfı B		D.Y. 5 Ürün 3 (x5) Proje 3 Küme 1 Sınıfı A		D.Y. 6
D.Y. 7 Ürün 2 (x1) Proje 2 Küme 1 Sınıfı A		D.Y. 8 Ürün 1 (x2) Proje 1 Küme 1 Sınıfı A		D.Y. 9
		D.Y. 10 Ürün 5 (x1) Proje 1 Küme 3 Sınıfı C		

Şekillerde her kutuda yazan D.A. depolama alanlarını temsil etmektedir.

Şekil 3. Kesin çözüm.

Figure 3. Exact solution.

Model oluşturulurken, amaçlardan biri kullanılan depolama yeri sayısını en aza indirmektir. Çıktıya göre hem kesin çözümün hem de sezgisel çözümün bu amaca uygun olduğu görülebilir. Her ikisinde de 10 depolama alanından 3'ü kullanılmamıştır.

Bir başka önemli kriter ise aynı projeye ait olan ürünleri birbirlerine yakın depolama yerlerine yerleştirmektir. Yukarıdaki şekillerde projeler mavi, sarı ve yeşil renkler ile vurgulanmıştır. Hem kesin çözüm hem de sezgisel çözümün analizine göre, kod bu açıdan iyi çalışmaktadır. İki çözümde de aynı projeye ait ürünlerin birbirlerine yakın yerleştikleri görülebilir. Örnek olarak ürün 4 & ürün 3, ürün 5 & ürün 1 ve ürün 2 & ürün 6 & ürün 7 verilebilir.

Diğer bir amaç aynı kümeye ait ürünleri birbirlerine yakın yerleştirmektir. Bu amacın çoğunlukla sağlandığı görülebilir. Sezgisel çözümde küme 2'ye ait ürünler birbirine yakinken küme 1 ve küme 3 ürünleri birbirinden daha ayrık yerleştirilmiştir. Kesin çözümde ise aynı şekilde küme 2'ye ait ürünler yakın konumlanmış ve küme 3 ürünleri birbirinden ayrı konumlanmıştır. Farklı olarak, küme 1 ürünleri sezgisel çözüme göre birbirine daha yakın konumlanmıştır.

Kısaca, kesin çözüm, sezgisel çözüme göre küme amacı açısından daha başarılı bir şekilde çalışmaktadır.

Başka bir kriter ise A-B-C sınıfı ürünleri atarken depolama yerleri ile G/Ç noktası arasındaki mesafeyi dikkate almaktır. Bu nedenle, G/Ç noktasına yakın depolama yerlerinin A sınıfına ait ürünleri tutması beklenmektedir. İki çözümde de A sınıfı ürünler olan ürün 1 & ürün 2 G/Ç noktasından uzak noktalara yerleştirilmiştir. C sınıfı olan ürün 5 ise her iki çözümde de G/Ç noktasından en uzak depolama alanına yerleştirilmiştir.

Son olarak, proje açısından kesin çözüm ve sezgisel çözümü kıyaslamak gerekirse, ikisi de aynı ve beklenen sonuçları

vermişlerdir. Diğer taraftan, kümeler açısından kıyasladığımızda kesin çözüm, kümeleri daha çok dikkate aldığı için daha başarılı olmuştur. Bu bölümün en başında kesin çözümde ve sezgisel çözümde de 7 ürün içinden 3 ürünün aynı depolama yerlerine yerleştirildiğini görebiliriz. Kesin çözüm ve sezgisel çözüm neredeyse aynı sonuçları vermiştir. Yalnız aynı kümeye ait ürünleri bir arada tutmak açısından kesin çözüm beklenildiği üzere daha başarılı sonuç vermiştir.

Sonuç olarak, aynı projeye ait ürünler birbirlerine yakın depolama yerlerine atanmıştır, aynı kümeye ait ürünlerin birbirlerine yakın depolama yerlerine atanması sağlanmıştır ve A-B-C sınıfı ürünleri atarken depolama yerleri ile G/Ç arasındaki mesafe göz önünde bulundurulmuştur. Bunların hepsi matematiksel modeldeki önceliklere (amaç fonksiyonu katsayıları) göre sağlanmaktadır. Ayrıca, tüm ürün tipleri her biri en az bir depolama yerine atandığından, ürünlerin hacminin hiçbiri, atanan depolama yerinin hacmini aşmadığından ve atanan ürün tipi sayısı, ürün tipi i'nin toplam miktarından büyük olmadığından modeldeki tüm kısıtlar açık bir şekilde karşılanmaktadır.

Kesin çözümün amaç fonksiyonu değeri 1,011 iken sezgisel çözümünki 1,061 olarak bulunmuştur (%4.7 göreceli boşluk). Sezgisel çözümün amaç fonksiyonu değeri daha büyük olmasına rağmen aradaki fark çok az olduğu için sezgisel çözümün kesin çözüme çok yaklaştığı ve ona başarılı bir alternatif olduğu sonucu çıkarılabilir. Ayrıca, sezgisel çözümün çözüm süresi kesin çözümün süresinden çok daha az olduğu düşünüldüğünde sezgisel çözüm kesin çözüme göre bu noktada daha iyi bir alternatif olmaktadır.

6 Hesapsal sonuçlar

6.1 Depolama yerleri mesafeleri

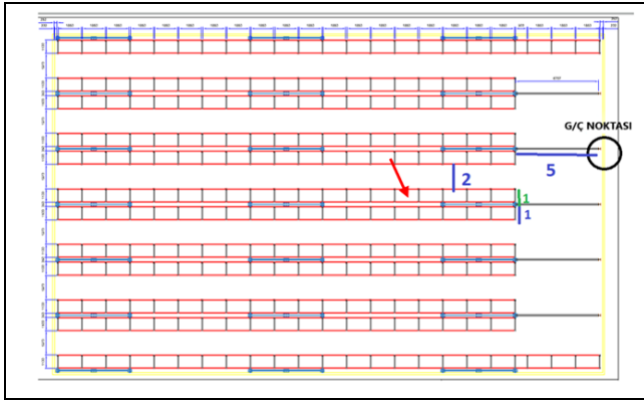
Çalışma boyunca yapılan en büyük veri düzenlemelerinden biri depolama yerleri arasındaki mesafeyi hesaplamaktır. Siloda toplam 4,179 adet depolama yeri bulunmaktadır. Bu da toplamda 17,646,041 adet değeri hesaplamak anlamına gelmektedir. İstenilen bu değerlere en hızlı yoldan ulaşmak için Floyd Warshall (FW) algoritması kullanılmıştır [22]. Bu algoritma girdi olarak bir mesafe matrisi alıp, ayrıt ağırlıklı yönlendirilmiş bir grafikte her düğüm noktası çifti arasındaki en kısa mesafeleri bulur. Önerilen bu algoritmanın uygulaması için Ryder (2020) tarafından Java'da yazılan kod heuristic algoritmasına entegre edilmiştir [23]. Öncelikle, Excel'de bir mesafe matrisi hazırlanmıştır. Bu matris hesaplanırken bazı varsayımlarda bulunulmuştur. İlk olarak yan yana ya da üst üste bulunan (direkt teması bulunan) depolama alan ikilileri arasında bir bağlantı olduğu kabul edilmiş ve bu bağlantıyı çift yönlü olarak sağlayan ayrıtların uzunluğu 1 m. olarak alınmıştır. Buradaki 1, her bir depolama yerinin boyutu olarak farz edilmiştir. Eğer depolama alanları arasında başka depolama alanı bulunuyorsa ve doğrudan bir bağlantı yoksa, bu bağlantıyı direkt olarak sağladığı varsayılan ayrıtların uzunluğu sonsuz olarak alınmıştır. Aynı mantık katlar arasında da uygulanmıştır.

Son olarak, aralarında koridor bulunan karşılıklı depolama alan çiftlerini bağlayan ayrıtların uzunluğu 3 m. olarak alınmıştır. Benzer şekilde bir koridorun iki tarafında bulunan ancak tam karşı karşıya olmayan depolama alanı çiftleri arasında direk bağlantı olmadığı ya da başka bir değişle aralarındaki ayrıt uzunluğunun sonsuz uzunlukta olduğu varsayılmıştır. Tüm bu bilgileri kullanarak 4,179 x 4,179 boyutunda bir matris elde edilmiştir. Floyd Warshall kodunu Java'da sezgisel koda

entegre ederek istenilen girdiye, depolama yerleri arasındaki mesafelere ulaşılmıştır.

6.2 Depolama yeri- G/Ç noktası mesafeleri

Değişiklik yapılması gereken ikinci veri seti ise depolama yerleri ve G/Ç noktası arasındaki mesafeleri hesaplamaktır. Bu mesafe hesaplaması, Manhattan uzaklığı prensibi kullanılarak hesaplanmıştır. Hesaplama, verilen iki nokta arasından geçen ve dik kesişen doğru parçalarının uzaklığı kadardır. Aynı depolama sütunundaki tüm depolama yerleri G/Ç noktasına eşit mesafede olduğundan, aynı sütunda farklı katlardaki tüm depolama yerleri eşit G/Ç noktası mesafe değerleri almıştır. Her bir koridorun 2 m., her bir depolama yerinin de 1 m. genişliğinde olduğu varsayılmıştır. Örneğin, Şekil 4'te kırmızı okla gösterilen depolama yerinin G/Ç noktasına uzaklığı 9 (yatay uzaklık) + 3 (dikey uzaklık)'ün toplanmasıyla 12 olarak bulunmuştur.



Şekil 4. Manhattan uzaklıkları.

Figure 4. Manhattan distances.

6.3 Ürün özellikleri

Bir başka veri düzenlemesi de ürünlerin hacimlerini hesaplamaktır. Fakat verilen veride bazı ürünlerin boyutları eksiktir. Veride ürünlerin kendi özelliklerine ait açıklamalar yer almaktadır. Bu ürün açıklamalarını kullanarak, boyutu eksik olan ürüne çok benzer diğer ürün açıklamalarına sahip ürünlerle eşleştirilmiştir. Yapılan bu eşleştirmeden sonra, bu boyutların ortalaması alınarak, boyutu eksik ürünün hacim hesaplamaları gerçekleştirilmiştir.

Son olarak, her bir ürünün miktarına ihtiyaç olmuştur ve bu bilgi veride düzenli bir şekilde bulunmamaktadır. Örneğin, bazı silo parçaları silonun stok bilgisinde gözükmemektedir ya da bir ürünü aynı lokasyonda birden çok görmek mümkün olmuştur çünkü depodan farklı zamanlarda toplanmaktadırlar. Stoklarda birden çok kez görülen bu ürünlerin miktarını gerçekte olduğundan daha yüksek miktarda hesaplama riskini ortadan kaldırmak için, ürünlerin miktarlarının ortalamaları alınıp bulunan değer ilgili ürünün miktarı olarak kullanılmıştır. Sadece silo yerleri içinde görülen, hiçbir tekrarı olmayan ürünler için ürünlerin miktarları doğrudan toplanmıştır.

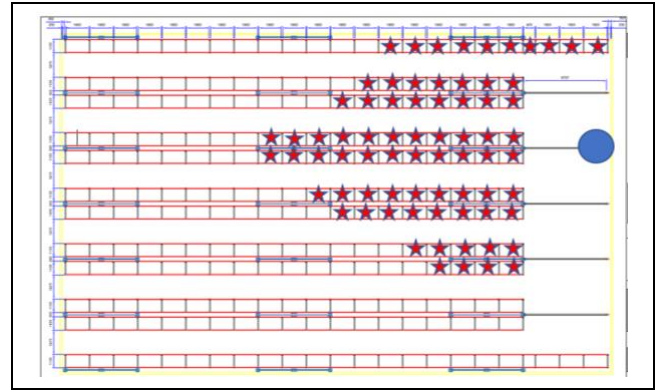
6.4 Gerçek veri ile çıktı

Sezgisel algoritma, gerçek veri kullanarak 2.50 GHz CPU ve 8 GB RAM'i olan bir bilgisayarda Windows işletim sisteminde Java kullanarak çalıştırılmıştır ve hesaplama zamanı 125 sn. olarak ölçülmüştür.

Çıktı Şekil 5'te görülmektedir. Mavi nokta G/Ç noktasını temsil etmektedir. Birbirlerine yakın depolama alanlarının

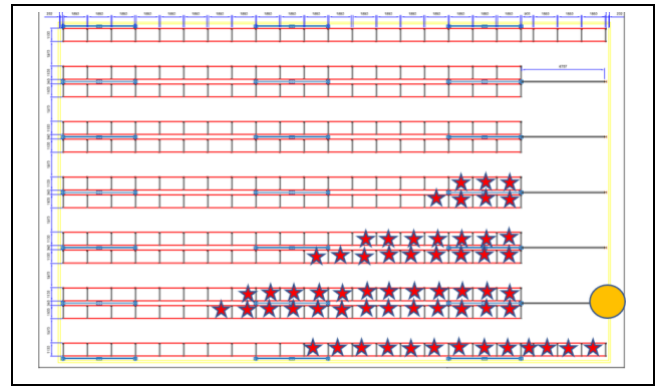
doldurulduğu görülebilir. Aynı zamanda, depolama alanı-G/Ç noktası mesafesi ve depolama alanlarının birbirlerine mesafelerinin dengelendiği de görülebilir. Çünkü ürünler hem G/Ç noktasına yakın hem de birbirlerine yakın konumlandırılmışlardır. Bu algoritmadaki "metrik" fikrinin doğru çalıştığını göstermektedir. Toplamda 127,189 ürün yerleştirilmiştir. Depolama alanı kullanımı olarak toplamda 4,179 depolama alanından 1,289 depolama alanı kullanılmıştır. Bu durum, silodaki depolama alanlarının yarısından fazlasının (2,890 depolama alanı) kullanılmadığı anlamına gelmektedir. Son olarak, farklı ürün tiplerinin aynı depolama alanına atanmasını engelleme kısıtının, hiçbir farklı ürün tipi aynı lokasyona atanmadığından, sağlanmış olduğu görülmektedir.

Başka bir senaryo olarak G/Ç noktasının mavi noktadan sarı noktanın olduğu yere alındığı durumda ürünlerin yerleşimi Şekil 6'da görülebilir. Bu senaryoda kullanılan depolama alanı sayısı 1,066 olmuştur ve bu kez 127,189 ürünün tamamı yerleştirilmiştir. Bu durumda açıkta kalan ürün olmamasının sebebi, yeni G/Ç noktasına yakın depolama yerlerinin hacimlerinin bir önceki duruma kıyasla daha büyük olmasıdır. G/Ç noktasının yerinin mavi noktadan sarı noktaya alındığı bu durumda, kullanılan depolama alanı sayısı 1,289'dan 1,066'ya düşmüştür. Bu çıktıya göre G/Ç noktasının yeni yeri, kullanılan depolama alanı sayısı azaldığından ve bütün ürünlerin yerleştirilebilir olmasından dolayı daha kullanışlı olmuştur denebilir.



Şekil 5. Çıktıya göre siloda dolu görünen depolama yerleri.

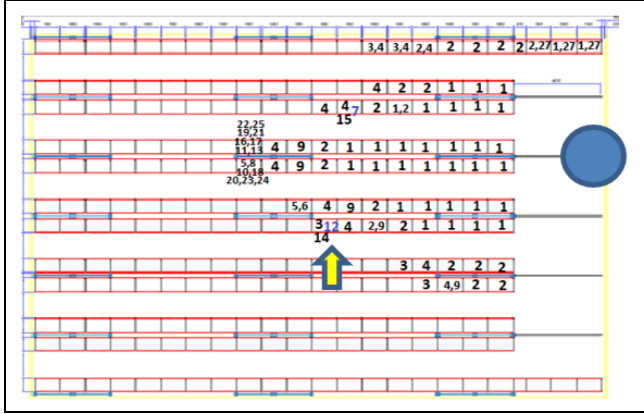
Figure 5. Filled slots in silo according to heuristic output.



Şekil 6. Değişen G/Ç noktası ile çıktıya göre siloda dolu görünen depolama yerleri.

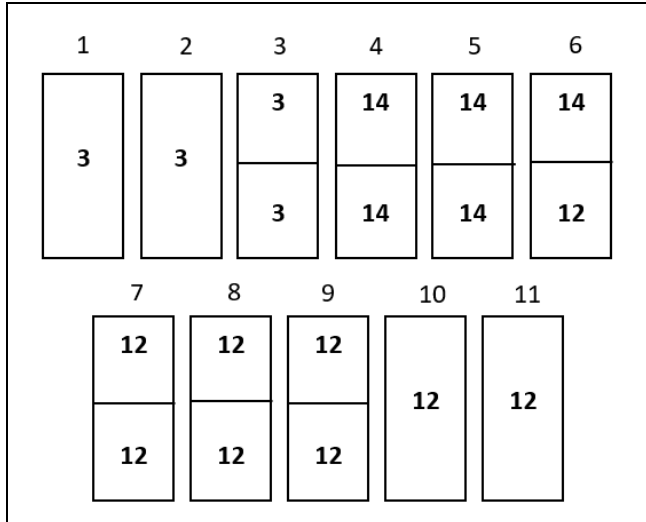
Figure 6. Filled slots in silo according to heuristic output with changed I/O point.

Çıktının projeler açısından daha detaylı analizi Şekil 7’de görülmektedir. Şekil 8 ise Şekil 7’de oklarla belirlenen depolama alanının yakından detaylı gösterimidir. Kutuların üzerindeki numaralar o raf sütunundaki kat numarasını, ikiye bölünmüş olan kutular ise o kattaki 2 farklı depolama gözünü göstermektedir.



Şekil 7. Proje gruplarının yerleşimi.

Figure 7. Project group allocation.



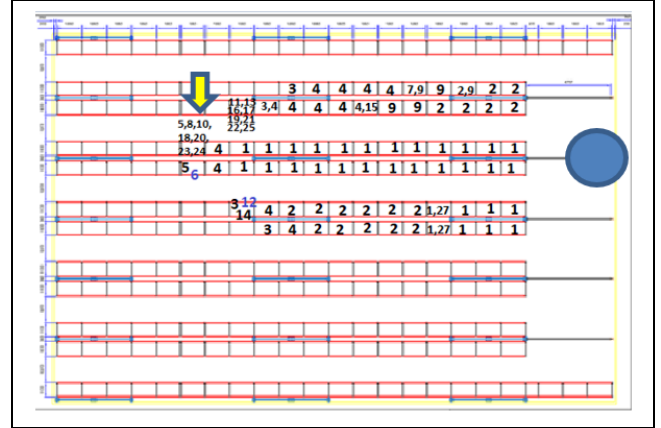
Şekil 8. Proje grupları yerleşimlerinin açılımı.

Figure 8. Opened form of project group allocation.

Şekil 7’de daha önce Şekil 5’te bahsedildiği gibi mavi nokta G/Ç noktasını temsil etmektedir. Şekil 7 ve Şekil 8’de aynı proje grubu ürünlerin başarılı bir şekilde birbirlerine yakın yerleştirildiği görülmektedir. Belirli bir proje grubuna yakın bir depolama gözünde yine aynı proje grubunun olduğu görülebilir. Ancak örnek olarak 2. ve 4. proje gruplarının nispeten dağınık yerleştiği görülmektedir. Bu problemde en önemli amaç proje gruplarını bir arada tutmayı sağlamak olduğu için, çözümü iyileştirmek adına sezgisel algoritmadaki metrik toplamının ilk kısmı 5 katsayısı ile çarpılmıştır. Yeni metrik ile elde edilen proje dağılımı çıktısı Şekil 9’da görülmektedir.

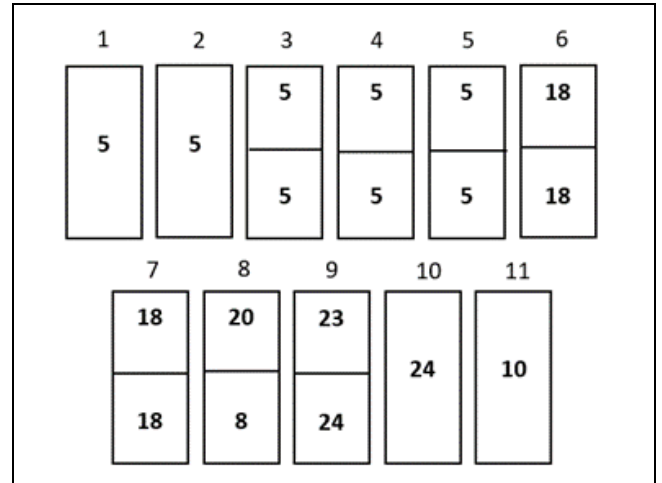
Şekil 9’da yeni metrik ile elde edilen proje dağılımının, Şekil 7’ye kıyasla daha başarılı olduğu görülmektedir, çünkü projeler daha toplu durmaktadır ve bütünlük daha fazla korunmuştur. Şekil 10 ise Şekil 9’da ok ile gösterilen depolama yerinin katlar bazında açılımıdır. Bu düzende 1,310 tane

depolama yeri kullanılmıştır. Eski metrik çıktısı ile aynı proje ürünü açıkta kalmıştır ve toplamda 127,188 ürün yerleşmiştir.



Şekil 9. Yeni metrik ile proje gruplarının yerleşimi.

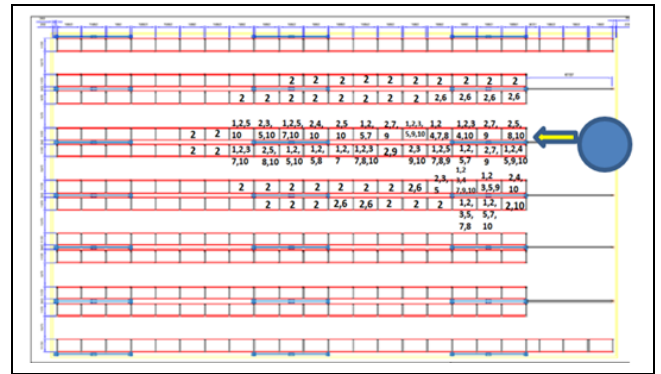
Figure 9. Project group allocation.



Şekil 10. Yeni metrik ile proje grupları yerleşimlerinin açılımı.

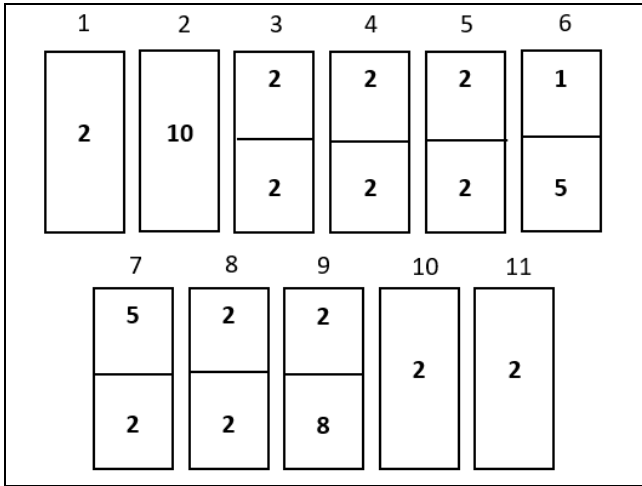
Figure 10. Opened form of project group allocation with the new metric.

Yeni metrik ile elde edilen çıktı, projelerin yerleşimi açısından daha başarılı olduğundan, bu metrik ile çıktılar güncellenmiştir. Şekil 11 ve Şekil 12’de kümelerin yerleşimini, Şekil 13 ve Şekil 14’te ise ABC sınıflarının yerleşimi görülmektedir.



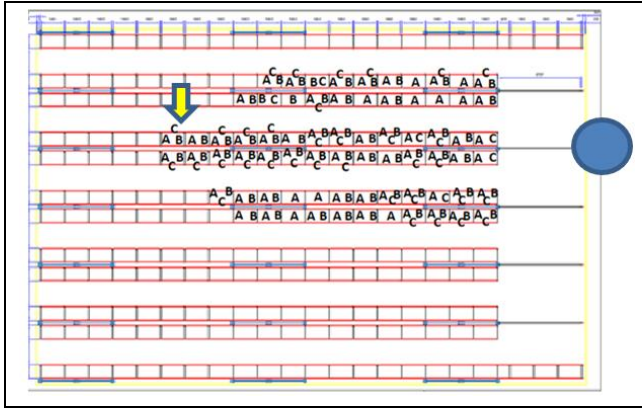
Şekil 11. Küme gruplarının yerleşimi.

Figure 11. Cluster group allocation.



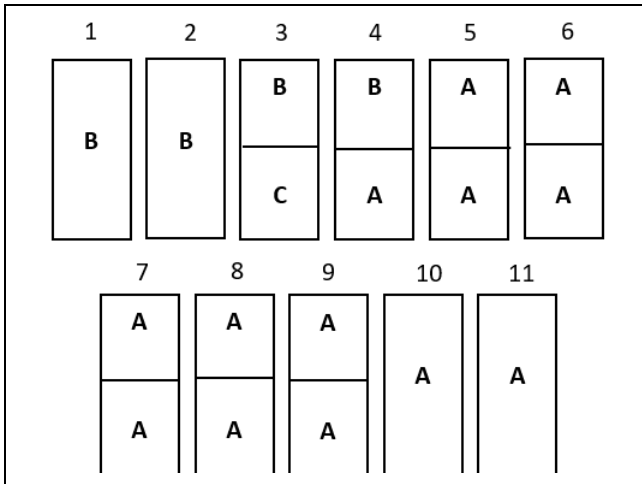
Şekil 12. Küme grupları yerleşimlerinin açılımı.

Figure 12. Opened form of cluster group allocation.



Şekil 13. ABC sınıflarının yerleşimi.

Figure 13. ABC class allocation.



Şekil 14. ABC sınıfları yerleşimlerinin açılımı.

Figure 14. Opened form of ABC class allocation.

Şekil 11 ve Şekil 12'de kümelerin yine oldukça başarılı bir şekilde yerleştirildikleri görülebilir. Şekil 13 ve Şekil 14'te A sınıfı ürünlerin G/Ç noktasına yakın konumlandırılmış olduğu görülmektedir. Fakat bazı C sınıfı ürünlerin de G/Ç noktasına çok yakın konumlandırıldığı ve bazı A sınıfı ürünlerin G/Ç noktasına en uzak noktalara yerleştirilmiş olduğu da

görülmektedir. Aynı proje grubu ürünleri, daha sonra da aynı kümeye ait ürünleri birbirlerine yakın yerleştirmek bu problemde sırasıyla en önemli 2 kriterdir. Onlardan sonra ise A, B, C sınıfı ürünlerin G/Ç noktasına olan mesafesi gelmektedir. Bu yüzden, beklendiği üzere, aynı projeye ve kümeye ait ürünleri birbirlerine yakın yerleştirmeye çalışırken A, B, C sınıflarının mesafeleri tam olarak sağlanamamıştır. Bu durum, sezgisel algoritmanın problemin amaçlarının öncelik sıralaması açısından doğru oluşturulduğunu göstermektedir.

6.5 Sistemdeki İyileşme

Bu çalışma için belirlenen performans kriterleri sipariş toplama süresi ve toplayıcıların ürünleri toplamak için kat ettiği toplam mesafe olarak belirlenmiştir. Bu bölümde çözüm, performans kriterleri bakımından incelenmiştir.

Sistemdeki iyileşmeyi görmek için ürünlerin yeni yerleşimlerine göre kullanılan depolama alanlarının G/Ç noktasına olan toplam mesafeleri, 17,500.11 birim bulunmuştur. Daha sonra bu değer, firmada güncel sisteme göre kullanılan depolama alanlarının G/Ç noktasına olan toplam mesafesi olan 34,996.5 birim ile karşılaştırılmıştır. Buradan, mesafelerde %49.99 oranında azalma olduğu görülmektedir.

Çözümü daha detaylı incelemek için 2019 Aralık'ın 14 günlük zaman dilimindeki verilerle çalışılmıştır. Her gün toplamda toplanan ürün sayısına bakılıp bu ürünlerin bulunduğu depolama alanları ile G/Ç noktası arasındaki toplam mesafeler hesaplanmıştır. Bu hesaplama hem firmanın güncel düzeni için hem de bu çalışmada elde edilen çözüm için yapılmıştır. Sonuçlar Tablo 5'te görülebilir.

Tablo 5. Karşılaştırma sonuçları.

Table 5. Comparison results.

Tarih	Toplanan ürün sayısı	Toplam G/Ç Mesafesi (yeni çözüm)	Toplam G/Ç Mesafesi (güncel sistem)
25-12-2019	9	118.18	169.92
24-12-2019	2	22.22	47.06
23-12-2019	17	183.15	350.25
22-12-2019	1	14.21	30.02
20-12-2019	26	309.85	594.72
19-12-2019	27	322.41	560.07
18-12-2019	39	512.15	788.83
17-12-2019	21	257.19	473.31
16-12-2019	8	101.04	158.81
13-12-2019	10	101.13	185.92
12-12-2019	33	428.36	710.10
11-12-2019	16	179.17	298.21
10-12-2019	12	125.68	240.91
9-12-2019	34	319.25	689.37

Önerilen çözüme göre toplam mesafe 2,993.97 birimdir. Firmanın güncel düzeni için ise bu değer 5,297.496 birimdir. Bu sonuçlar performans kriterleri ile ilişkilendirilecek olursa, önerilen çözümlerle sistemde %43.48 oranında iyileşme olmaktadır.

Ayrıca, kullanılan depolama alanı sayısı firmanın güncel durumunda 2508 iken bu sayı önerilen çözüm ile 1289'a düşmüştür. Performans kriterleri ile ilişkilendirildiğinde, önerilen çözümle sistemde %48,6 oranında iyileşme olmaktadır.

7 Sonuç

Depo operasyonları tedarik zincirinde önemli bir rol oynamaktadır. Firmalar, depo gereksinimlerini analiz edip deponun mümkün olan en iyi kullanımını belirlemelidirler. Depoların verimli yönetimi depoda daha fazla boş alan ve zamandan tasarruf edilmesini sağlar.

Bu çalışmada, sezgisel yaklaşım kullanılarak toplanma sıklıklarının dikkate alan veri güdümlü depolama yeri atama problemi sunulmuştur. Çalışmanın sonunda, depo için yeni bir yerleşim düzeni elde edilmiştir. Bu plan, ürünlerin atandığı depolama yerlerini ve atanan miktar bilgisini içermektedir. Bu çalışmada amaç, aynı proje ve küme grubu ürünlerinin birbirlerine olan mesafesi ve A, B, C sınıfı ürünlerin G/Ç noktasına olan mesafesi dikkate alınarak kullanılan depolama yerleri sayısını azaltmaktır. Önerilen çözüm yöntemi sayesinde, mevcut sistemde %49.99'a varan iyileşme gözlemlenmiştir.

Bu çalışma ileride raf doluluğunu gözeterek genişletilebilir. Bu, sadece depolama yerleri ve G/Ç noktası arasındaki mesafeyi dikkate alarak değil, aynı zamanda depolama yerlerinin hacimlerini de dikkate alarak sağlanabilir. Atanacak bir sonraki ürün için ürünün hacmi ve depolama yerlerinin hacmi, küçük hacimli ürünler diğer depolama yerlerine kıyasla daha küçük hacimli depolama yerlerine atanacak şekilde dikkate alınabilir. Sağlanamayan raf doluluğu, zaman içinde depolama yerlerinin kullanım dışı kalmasından dolayı atanacak ürünlerin kendi proje veya küme grubundan alakasız lokasyonlara atanmasına yol açabileceğinden, depo kullanımı açısından önemli bir husustur.

Buna ek olarak, bu çalışmada kullanılan ağgözlü algoritma başlangıç çözümü olarak kabul edilip, en yakın komşu algoritması, tavlama benzetimi algoritması ve genetik algoritma gibi eniyileme sezgiselleri kullanılarak çalışma daha da genişletilebilir.

Bu çalışmada, depo boş olarak kabul edilip ürünler sıfırdan yerleştirilmiştir. Bu çözüm gerçek zamanlı bir çözüm değildir. Gerçek zamanlı bir çözümde yerleşik bir depoda, depoya yeni gelen/çıkan ürünlerin yerleşimi gerekmektedir. Bu yüzden bu gereksinimi karşılayacak gerçek zamanlı bir algoritma geliştirilip, kullanıcı dostu bir arayüz ile kullanıcıya sunulabilir.

8 Conclusion

Warehouse operations take a key role in the supply chain. Companies need to analyze their warehousing needs and then determine the best possible usage of the warehouse. Therefore, an effective management of the warehouse enables the company to have more free spaces and save time.

Throughout this paper, we introduced a data driven storage location assignment problem considering order picking frequencies using a heuristic approach. At the end of this study, a new allocation plan is obtained for the warehouse. This plan gives the information of which item is located to which slot and how many of that item is allocated. During this study, the goal was to minimize the number of slots used while paying attention to the distance between items which belong to the same project, the distance between parts who are frequently picked together and the distance between A, B or C classed

items locations to the particular I/O point. With the suggested solution, an improvement of up to 43% in the current system is observed.

This problem can be extended in a way that slot fullness is satisfied. It can be achieved by considering not only the distances between slots and the I/O point, but also the volumes of slots. According to the next item's volume to be allocated, the volume of the slots can also be considered in a way that items which have small volume are assigned to slots which have smaller volume in comparison with other slots. It is an important aspect of warehouse usage since unsatisfied slot fullness may result in unavailability of slots in time for certain items, and the items would be allocated to an irrelevant location with their project or cluster groups.

In addition, the greedy algorithm, which is used in this study, can be considered as the initial solution and the work can be further expanded by using improvement heuristics such as nearest neighborhood algorithm, simulation annealing algorithm and genetic algorithm.

In this study, the warehouse was considered as empty and the items were allocated to slots in this setting. This planning approach can be considered as an offline approach. In a real-time setting, allocating the incoming/outgoing items in an already settled warehouse is required. Hence, an online algorithm that fulfills this requirement and embedded in a user interface can be developed.

9 Yazar katkı beyanı

İpek ÇOBANOĞLU, ve İrem GÜRE literatür taraması, sistemin incelenmesi ve veri toplanmasında, modelin GAMS programlama dilinde kodlanmasında ve koşturulmasında, test sonuçlarının analiz edilmesinde ve makalenin yazımında, Vedat BAYRAM ise fikrin oluşması, problem tanımının yapılması, optimizasyon modelinin hazırlanması ve yazım denetimi ve içerik açısından makalenin kontrol edilmesi başlıklarında katkı sunmuşlardır.

10 Etik kurul onayı ve çıkar çatışması beyanı

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur. Hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır.

11 Kaynakça

- [1] Frazelle EH. Stock Location Assignment and Order Picking Productivity. Doctoral Dissertation, Georgia Institute of Technology. Atlanta, United States, 1989.
- [2] Kofler M. Optimising the Storage Location Assignment Problem Under Dynamic Conditions. Doctoral Dissertation, Technisch-Naturwissenschaftliche Fakultät, Linz, Austria, 2014.
- [3] Gu J, Goetschalckx M, McGinnis LF. "Solving the forward-reserve allocation problem in warehouse order picking systems". *Journal of the Operational Research Society*, 61(6), 1013-1021, 2010.
- [4] Kofler M, Beham A, Wagner S, Affenzeller M, Achleitner W. "Re-Warehousing vs. Healing: Strategies for warehouse storage location assignment". *3rd IEEE International Symposium on Logistics and Industrial Informatics (LINDI)*, Budapest, Hungary, 25-27 August 2011.
- [5] Frazelle E, Sharp G. "Correlated assignment strategy can improve order-picking operation". *Industrial Engineering*, 4, 33-37, 1989.

- [6] Muppani VR, Adil GK. "Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach". *Omega The International Journal of Management Science*, 36(2008), 609-618, 2007.
- [7] Battista C, Fumi A, Giordano F, Schiraldi MM. "Storage Location Assignment Problem: implementation in a warehouse design optimization tool". *Proceedings of the Conference Breaking Down the Barriers Between Research and Industry*. Padua, Italy, 14-16 September 2011.
- [8] Yang C, Nguyen TPQ. "Constrained clustering method for class-based storage location assignment in warehouse". *Industrial Management & Data Systems*, 116(4), 667-689, 2015.
- [9] Jane CC, Laih YW. "A clustering algorithm for item assignment in a synchronized zone order picking system". *European Journal of Operational Research*, 166(2005), 489-496, 2005.
- [10] Bindi F, Manzini R, Pareschi A, Regattieri A. "Similarity-based storage allocation rules in an order picking system: an application to the food service industry". *International Journal of Logistics: Research and Applications*, 12(4), 233-247, 2009.
- [11] Ene S, Öztürk N. "Storage location assignment and order picking optimization in the automotive industry". *The International Journal of Advanced Manufacturing Technology*, 60(2012), 787-797, 2011.
- [12] Norén P, Eriksson J. Lund Institute of Technology. Department of Industrial Management and Engineering. "A heuristic algorithm for space allocation in a pallet storage warehouse". <https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=1883111&fileId=1883113> (12.06.2020).
- [13] Dantzig, GB. "Discrete variable extremum problems". *Operations Research*, 5(2), 266-288, 1957.
- [14] Xu J, Lim A, Shen C, Li H. "A heuristic method for online warehouse storage assignment problem". *IEEE International Conference on Service Operations and Logistics, and Informatics*, Beijing, China, 12-15 October 2008.
- [15] Wisittipanich W, Kasemset C. "Metaheuristics for warehouse storage location assignment problems". *Chiang Mai University Journal of Natural Sciences*, 14(4), 361-377, 2015.
- [16] Kim, J, Méndez, F, Jimenez, J. "Storage location assignment heuristics based on slot selection and frequent itemset grouping for large distribution centers". *IEEE Access*, 8, 189025-189035, 2020.
- [17] Yongxia Z. "On a multi-standard ABC analysis method in the inventory management of small and medium-sized enterprises". *International Conference on Advanced Information and Communication Technology for Education (ICAICTE)*, Hainan, China, 20-22 September 2013.
- [18] R Core Team. "R: A language and environment for statistical computing". R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org/> (16.03.2013).
- [19] Wickham H. "Reshaping data with the reshape package". *Journal of Statistical Software*, 2007. <https://doi.org/10.18637/jss.v021.i12>.
- [20] Guerriero, F, Pisacane, B, Rende, F. "Comparing heuristics for the product allocation problem in multi-level warehouses under compatibility constraints". *Applied Mathematical Modelling*, 39, 7375-7389, 2015.
- [21] Durmuş B, İşçi Güneri Ö, İncekırık A. "Comparison of classic and greedy heuristic algorithm results in integer programming: Knapsack problems". *Mugla Journal of Science and Technology*, 5(1), 34-42, 2019.
- [22] Floyd Robert W. "Algorithm 97: Shortest path". *Communications of the ACM*, 5(6), 344-348, 1962.
- [23] Ryder A. "Floyd-Warshall Algorithm: Shortest Path Between All Pair of Nodes". <https://iq.opengenus.org/floyd-warshall-algorithm-shortest-path-between-all-pair-of-nodes/> (05.05.2020).