

Süpermarket Yerleşim Problemi İçin Tavlama Benzetimi Algoritması Yaklaşımı

Mevlüt UYSAL^{1*}, Uğur ÖZCAN²

¹Gazi Üniversitesi, Bilişim Enstitüsü, Yönetim Bilişim Sistemleri, Ankara, Türkiye

²Gazi Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Ankara, Türkiye

Geliş Tarihi: 12.01.2019

Kabul Tarihi: 14.06.2019

*Sorumlu Yazar: mevlutuysal@gazi.edu.tr

Öz

Modern üretim sistemlerinde, sürekli artan ürün çeşitliliği, karışık modellenmiş montaj hatlarını gerekli parçalarla beslemek için kullanılan iç lojistik sistemleri için büyük bir zorluk teşkil etmektedir. Bu soruna bir cevap olarak, özellikle otomotiv endüstrisinden birçok üretici, esnek ve güvenilir parça tedariki sağlamak için süpermarket konseptini benimsemiştir. Bu bağlamda süpermarket, parçaların montaj hattına yakın olarak saklandığı ve küçük çekici trenler aracılığıyla istasyonlara taşındığı merkezi olmayan lojistik merkezidir. Bu çalışma kapsamında işletme zeminine kurulacak süpermarketlerin sayısının ve yerlerinin belirlenmesi problemi ele alınmıştır. Daha önceki çalışmalarda problemin çözümü için dinamik programlama ve genetik algoritma gibi yöntemler kullanılmıştır. Komşuluk yapısı içeren algoritmaların problemin çözümünde etkili olabileceği düşünülerek, çalışmamızda bir tavlama benzetimi algoritması geliştirilmiş ve algoritma bir dizi test problemi üzerinde test edilmiştir. Testler sonucunda, geliştirilen algoritmanın düşük çalışma sürelerinde başarılı sonuçlar verdiği görülmüştür.

Anahtar Kelimeler: Karışık modellenmiş montaj hatları, İç lojistik, Süpermarket, Tavlama benzetimi

A Simulated Annealing Algorithm Approach for Supermarket Location Problem

Abstract

In modern production systems, the ever-increasing product variety is a major challenge for the in-house logistics systems used to feed mixed-model assembly lines with the necessary components. In response to this problem, many manufacturers from the automobile industry have adopted supermarket concept to provide flexible and reliable part feeding. In this context, supermarket is a decentralized logistics center where the parts are stored close to the assembly line and transported to the stations via small tow trains. Within the scope of this study, the problem of determining the number and location of supermarkets to be established on the operational floor has been discussed. In previous studies, methods such as dynamic programming and genetic algorithm have been used to solve the problem. Considering that neighboring algorithms can be effective in solving the problem, an annealing simulation algorithm was developed in our study and the algorithm was tested on a number of test problems. As a result of the tests, it was found that the developed algorithm gave successful results in low running times.

Keywords: Mixed-model assembly lines, In-house logistics, Supermarket, Simulated annealing algorithm

1. Giriş

Günümüz rekabetçi ortamında firmaların etkin ve kaliteli ürünleri üretip müşterilerine istedikleri zaman ve istedikleri miktarda ulaştırmaları büyük önem taşımaktadır. Hız, kalite ve düşük maliyet beklentisi içinde bulunan müşteriler olduğu sürece, mükemmel lojistik performans mutlaka yeni pazarlar oluşturacaktır. Bu özelliklerin olması gerektiği gibi sağlanabilmesi için üretimi gerçekleştirilen ürünlerin iç lojistik faaliyetlerinin de etkin bir yapıda gerçekleştirilmesi gerekmektedir. Tam Zamanında Üretim (TZÜ) felsefesine göre montaj hattına, iç lojistik faaliyetleri kapsamında yer alan, montaj işlemleri için gerekli malzeme ve parça tedarikinin istenen zamanda istenen miktarda ve istenen yere sağlanması büyük önem taşımaktadır. Özellikle montaj hattının malzeme ve parça yoksunluğundan dolayı durması yüksek maliyetlerin ortaya çıkmasına neden olmaktadır.

Son yıllarda üreticiler tesis içinde hızlı, esnek ve güvenilir parça lojistiği sağlayabilmek için süpermarket konseptini benimsemişlerdir. TZÜ süpermarketi yakınındaki montaj hattının parça talebini karşılayabilmek için kurulan, merkezi olmayan ara bir saklama alanıdır. Süpermarketler merkezi depodan montaj hatlarının ihtiyaç duyduğu farklı büyüklükteki montaj bileşenleri ile doldurulur ve küçük kutular halinde çekici trenler aracılığıyla servis edilir. Kısa mesafeler ve küçük çaplı teslimatlar sayesinde parça tedariki daha esnek bir şekilde yönetilebilir ve öngörülemez durumlarla daha kolay başa çıkılabilir (Battini ve ark., 2015). Süpermarketler tesis içerisinde montaj hattına yakın alanlarda yer alır ve küçük de olsa bir alan işgal ederler, eğer mevcut tesis içerisinde yeni süpermarketlerin oluşturulması gerekiyorsa burada tesis içi yerleşim problemi ortaya çıkar.

Yerleşim problemi fabrika zemini üzerinde kurulacak süpermarket alanlarının sayısının ve yerleşiminin belirlenmesidir. Fabrika alanı çok değerlidir, bu nedenle çok sayıda süpermarket yaratmak genellikle avantajdan daha fazla maliyete neden olacaktır. Diğer taraftan, az sayıdaki, kötü yerleştirilmiş süpermarketlere sahip olmak olumlu etkilerini büyük ölçüde azaltmasına rağmen, merkezileştirilmiş depolarla kurulan taşıma sistemlerinden daha iyi sonuçlar verir. Bu nedenle mümkün olan alanı kullanarak ve mümkün olan en iyi süpermarket sayısını seçerek ve onları stratejik olarak yerleştirmek kritik bir öneme sahiptir (Emde ve Boysen, 2012).

Merkezi olmayan parça lojistiği ve merkezi olmayan depo konseptleri literatürde geniş yer bulmuştur. Örneğin Yano ve ark. (1998) merkezi olmayan lojistiğin avantajlarını vurgularken, Wanke ve Zinn (2004) merkezi olmayan depo kullanımındaki stratejik karar seçeneklerini incelemiştir. Bunlara rağmen literatürün çoğu bu kararı alırken üretim ortamından daha çok tedarik zinciri ortamına yoğunlaşmıştır. Az sayıda çalışmada üretim ortamındaki merkezi olmayan depo (süpermarket) konseptine yer verilmiştir.

Battini ve ark. (2010) siparişe dayalı montaj ortamında merkezi olmayan depo karar seçeneklerini incelemiştir. Montaj hatlarını süpermarketlere bileşen ortaklığına göre atamışlar ve süpermarketleri mümkün olduğunca kendi hatlarına yakın olacak şekilde yerleştirmişleridir. Her süpermarkette tutulacak güvenlik stokları nedeniyle oluşacak stok maliyetlerini ve montaj hattının süpermarketten yeniden doldurulması nedeniyle doğacak taşıma maliyetlerini ölçmek için bir yöntem önermişlerdir.

Faccio ve ark. (2013) tarafından yapılan çalışmada merkezi olmayan depolarda malzeme ve parça tedariki ele alınmıştır. Taşıyıcılar daha önce istasyonlara bıraktıkları kutuları tekrar toplamakta ve ihtiyaç duyulan malzeme ve parçalar ile doldurarak tekrar istasyonlara taşımaktadırlar. Kanban sayısı minimizasyonu altında karışık modellenmiş montaj hatları dikkate alınmıştır.

Battini ve ark. (2013), merkezi olmayan depo (süpermarket) konseptini ele almışlardır ve süpermarket konseptinde ortaya çıkan temel problemleri tanımlamışlardır. Bununla birlikte konu ile ilgili detaylı bir literatür araştırması sunmuşlardır.

Boysen ve ark. (2015) otomotiv endüstrisindeki parça lojistiğinin temel süreçlerini siparişlerin alınmasından boş parça konteynırlarının geri dönüşüne kadar ele almışlar ve konu ile ilgili detaylı bir literatür araştırması sunmuşlardır. Karşılaşılan karar problemlerini açıklamışlar ve süpermarket konseptinin avantajlarından bahsetmişlerdir.

Az sayıda çalışma süpermarket yerleşim problemine (SYP) çözüm önerisi getirmiştir. Bunlardan Emde ve Boysen (2012), tek bir montaj hattı boyunca istasyonlara parça tedariki sağlayan süpermarketlerin yerleştirilmesi için optimal bir algoritma sunmuşlardır. Yeni bir süpermarket eklemenin yaratacağı sabit maliyet ile istasyon talepleri ile ağırlıklandırılmış taşıma mesafelerinin azalması arasındaki farkı hesaplayan ve ortaya çıkan problemi polinom zamanda çözen bir dinamik programlama algoritması modellemişlerdir. Alnahhal ve Noche (2015) ise aynı problemi tam sayılı programlama modeli ve genetik algoritma kullanarak çözmeye çalışmışlardır. Emde ve Boysen'in (2012) çalışmasından farklı olarak, problemi çözerken tesis içinde süpermarket kurulumu için uygun olmayan bölgeleri göz önüne alınmıştır. Nourmohammadi ve ark. (2018) ise süpermarket yerleşim problemini değişken istasyon talepleri ile birlikte incelemişlerdir. Değişken istasyon talepleri, güvenlik stoğu ve kurulum maliyetlerini de içeren bir SYP modeli önermişler ve önerdikleri modeli test problemleri ve bir işletme üzerinde denemişlerdir.

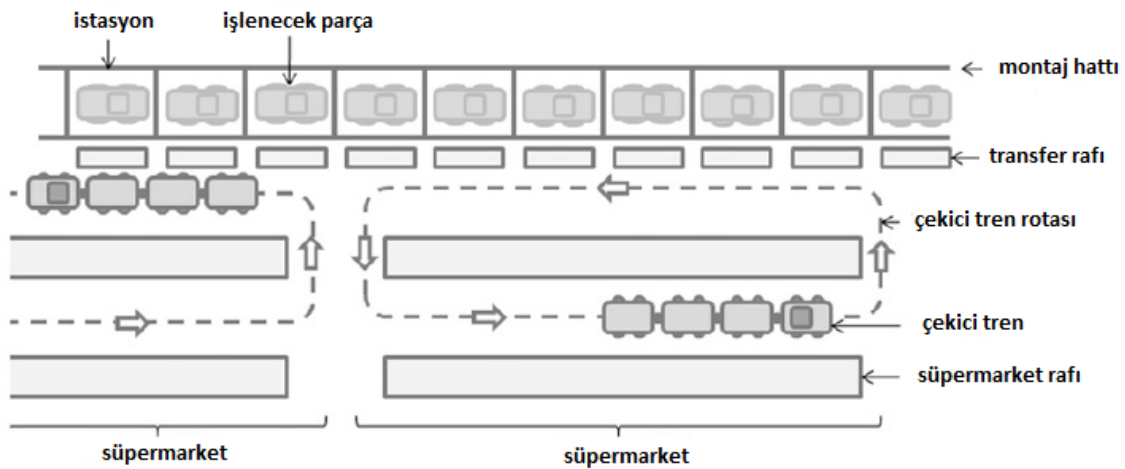
Literatürde daha önce kullanılmadığı için SYP çözümünde komşuluk yapısı içeren algoritmalarının etkili sonuçlar verebileceği düşünülmüştür. Bu sebeple çalışmamızda SYP için, literatürde farklı alanlarda yaygın olarak kullanılan, komşuluk yapısı içeren algoritmalarından tavlama benzetimi (TB) algoritmasının kullanılması uygun görülmüştür. Emde ve Boysen (2012)'nin çalışmasında kullanılan matematiksel modelden faydalanılarak bir TB algoritması geliştirilmiştir. Geliştirilen algoritma bir dizi test problemi ile test edilmiştir. Sonuçların başarısını ölçmek için

birleme yöntemiyle test problemlerinin optimal sonuçları edilmiş ve geliştirilen algoritmanın sonuçları ile karşılaştırılmıştır.

2. Materyal ve Metot

2.1. Problem Tanımı

Karışık modellenli montaj hatlarında süpermarket yerleşim problemi çözülürken karar verilmesi gereken değişkenler vardır. Her montaj istasyonunun (s) hangi süpermarketten kaç kutu parça (d_s) alması gerektiği belirlenmelidir. Montaj istasyonlarının ve aynı zamanda çekicilerin durma noktalarının fabrika zemini üzerinde koordinatları (a_s, b_s) hesaplanır. Bu koordinatlar sayesinde istasyon s ve $s+1$ arasındaki mesafe (e_s) ve aynı zamanda çekici tarafından ziyaret edilecek s istasyonu ile i süpermarketi arasındaki mesafeleri (z_{is}) metrik olarak kolaylıkla hesaplanabilir. Sonrasında SYP şu şekilde tanımlanabilir: Optimum süpermarket sayısı nedir ve hangi süpermarket hangi istasyonlara hizmet vermelidir? Bu sorular cevaplandıktan sonra süpermarket pozisyonları hesaplanır. Şekil 1'de iki süpermarketi olan bir montaj hattı düzeni gösterilmiştir (Emde ve Boysen, 2012).



Şekil 1. İki süpermarketi olan bir fabrika zemini

Çekici trenler her bir istasyona ayrı ayrı uğramaktansa birden çok duruş noktasıyla bir belirli bir rota üzerinde hareket eder. Bu sebeple bir süpermarketten istasyona giden yol düz bir çizgi değildir, bunun yerine çekici trenin toplam dolaşacağı mesafe hesaplanmalıdır. Çekici tren önce rotasındaki ilk istasyona, sonra sırayla her bir istasyona ve sonunda yeniden yüklenmek üzere süpermarkete uğramalıdır. Bu durumda bile mesafeleri azaltmak her zaman iyi çözümleri garanti etmez. Çekicilerin belirli bir kapasitesi olduğundan, bir turda kaç tane ve hangi istasyonlara hizmet edileceği istasyonların parça talebine bağlıdır. Yüksek parça talebi olan istasyonların aynı rotada

olması durumunda, çekici tüm istasyonların ihtiyacını karşılayamaz. Bundan dolayı uzaklıklar süpermarketin hizmet verdiği istasyonların toplam talebi ile ağırlıklandırılmalıdır. Son olarak, üçüncü bir bileşen olarak, bir süpermarket oluşturmak ve sürdürmek için sabit maliyet de hesaba katılmalıdır. Bunları göz önüne alarak Süpermarket Yerleşim Problemi aşağıdaki gibi tanımlanabilir (Emde ve Boysen, 2012):

$s = 1, \dots, S$ süpermarket tarafından hizmet verilecek ardışık istasyonlar olsun. SYP bu istasyonların birbirinden ayrı bir süpermarket tarafından hizmet verilen, değişken alt gruplara bölünmesinden oluşur. Bir çözüm $X(n) = \{1, x_2, x_3, \dots, x_n, S + 1\} \rightarrow \{2, \dots, S\}$ şeklinde, x_i 'in süpermarketin hizmet verdiği istasyonların en solundakini ve $x_{i+1}-1$ 'in en sağdaki gösterdiği bir vektör olarak gösterilebilir. Tüm istasyonların bir süpermarket tarafından hizmet verilmesi gerektiği için ilk süpermarketin hizmet alanının en solundaki istasyon her zaman 1 ve sonuncunun en sağdaki istasyon da her zaman S 'dir. Çözüm vektörün uzunluğu $n+1$ 'dir ve elemanları aşağıdaki amaç fonksiyonu ve kısıtlara uymalıdır (Emde ve Boysen, 2012).

$$\text{Min } F(X(n)) = \sum_{i=1}^n \sum_{s=x_i}^{x_{i+1}-1} d_s \cdot (\sum_{\tau=x_i}^{x_{i+1}-2} e_{\tau} + z_{i,x_i} + z_{i,x_{i+1}-1}) + n \cdot \Gamma \quad (1)$$

$$\text{S.T. } x_{i+1} \geq x_i + 1 \quad \forall i = 1, \dots, n \quad (2)$$

$$z_{is} = |a_{x_i} + (a_{x_{i+1}-1} - a_{x_i})/2 - a_s| \\ + |b_{x_i} + (b_{x_{i+1}-1} - b_{x_i})/2 - b_s| \quad \forall i = 1, \dots, n; \quad s = 1, \dots, S \quad (3)$$

Amaç fonksiyonu (1) n sayıdaki Γ ile ağırlıklandırılmış süpermarketleri ve süpermarketin hizmet alınındaki istasyonların toplam talepleri ile çekicinin dolaşacağı toplam mesafelerin çarpımlarının toplamını minimize etmeyi amaçlamaktadır. Kısıt (2) hizmet alanlarının çakışmamasını sağlamaktadır. Kısıt (3) ise her bir süpermarketten her bir istasyona olan mesafeyi hesaplamaktadır. Fabrika zeminleri genellikle çizgi halinde sürüş yolları ve keskin dönüşlerle tasarlandığı için mesafeleri ölçmek için kullanılacak en iyi yöntem Manhattan metriğidir. Çekiciler bir süpermarketten başlayarak tüm istasyonları uğradıktan sonra tekrar süpermarkete döneceği için, süpermarketler hizmet ettiği istasyonların başlangıç ve bitiş noktalarının ortasına yerleştirilir. Diğer yandan, bir montaj hattıyla çakışma ihtimalinden dolayı süpermarketler her zaman istasyonların tam ortasına yerleştirilemediği durumlar olabilir. Öyle olsa bile, süpermarketler her zaman istasyonlara eşit uzaklıkta ve belirli bir mesafede yerleştirilmesi gerektiği için bu durumun ayrıca modellenmesine gerek yoktur (Emde ve Boysen, 2012).

Tablo 1. Kısaltmaların gösterimi

S	İstasyon sayısı ($s = 1, \dots, S$)
n	Süpermarket sayısı gösteren değişken ($i = 1, \dots, n$)
r	Her süpermarket için sabit maliyet
e_s	s istasyonun s+1 istasyonuna olan mesafe
d_s	s istasyonunda her vardiyada tahmin edilen kutu talebi
a_s	s istasyonunun x koordinatı
b_s	s istasyonunun y koordinatı
x_i	i süpermarketi tarafından hizmet verilen ilk istasyon
z_{is}	i süpermarketinden s istasyonuna olan mesafe

2.2. Tavlama Benzetimi Yaklaşımı

SA, yinelemeli rastgele bir arama tekniğidir ve çeşitli kombinasyonel optimizasyon problemlerini çözmek için yaygın olarak kullanılmaktadır. SA ilk olarak Metropolis, Rosenbluth ve Teller (1953) tarafından metallerin tavlama işlemini simüle etme amacıyla önerilmiştir ve Kirkpatrick, Gelatt ve Vecchi (1983) tarafından yinelemeli bir optimizasyon yöntemi olarak tanıtılmıştır.

TB algoritması önceden belirlenen bir başlangıç sıcaklığı ile başlar (T_0). Bir başlangıç çözümü oluşturulur (S_0) ve bu aynı zamanda geçerli çözüm (S_C) ve ilk en iyi çözüm olur (S_B). Başlangıç çözümünün amaç fonksiyonu değeri (E_0), geçerli çözüm (E_C) ve en iyi çözümün amaç fonksiyonu değeri olur (E_B). Geçerli çözümden (S_C) bir değişiklik ile komşu çözüm (S_N) oluşturulur ve amaç fonksiyon değerleri hesaplanır (ΔE). Eğer komşu çözümün amaç fonksiyonu değeri (E_N), E_C 'den daha iyiye, yeni geçerli çözüm olarak kabul edilir. Ancak E_N E_C 'den daha kötüye, $\exp(-\Delta E/T)$ (T geçerli sıcaklık) olasılığı ile S_N yeni S_C olarak kabul edilir, diğer durumda S_N reddedilir. T parametresi bir soğutma fonksiyonu ($T = \alpha \times T$) ile yavaşça azaltılır ve T_{min} ulaşılacak son sıcaklıktır. TB algoritmasının kötü çözümleri kabul ederek yerel optimum noktalardan kurtulma kapasitesi vardır (Özcan ve Toklu, 2009).

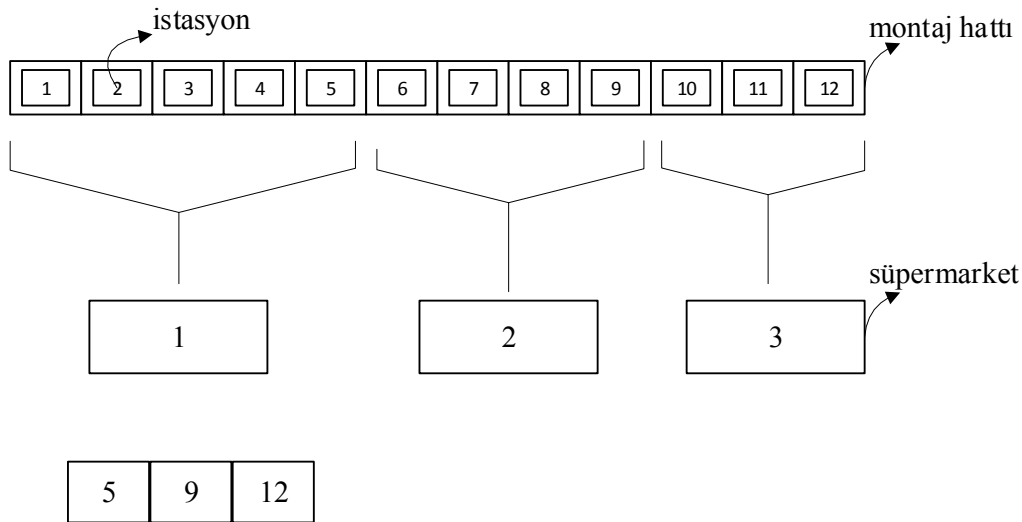
Çalışmada önerilen TB algoritması aşağıdaki gibidir:

Prosedür: TB

1. Giriş parametreleri: $T_0, T_{min}, \alpha, T = T_0$.
2. Bir S_0 üret ve E_0 'i hesapla.
3. $S_C = S_0, S_B = S_0, E_C = E_0, E_B = E_0$.
4. S_C 'den bir S_N üret ve E_N 'i hesapla
5. $\Delta E = E_N - E_C$.
6. Eğer $\Delta E \leq 0$ komşu çözümü geçerli çözüm olarak kabul et, $S_C = S_N, E_C = E_N$ ve Adım 9'a git.
7. Eğer $\Delta E > 0$ rnd isminde $[0, 1]$ arasında bir değişken üret.

8. Eğer $\exp(-\Delta E/T) > \text{rnd}$ ise komşu çözümü geçerli çözüm olarak kabul et, $S_C = S_N$, $E_C = E_N$; diğer durumda, S_C ve E_C değişmeden kalır ve Adım 10'a git.
9. Eğer $E_C < E_B$ ise $S_B = S_C$; $E_B = E_C$; diğer durumda, S_B ve E_B değişmeden kalır.
10. $T = T \times \alpha$.
11. Eğer $T \geq T_{\min}$ ise Adım 4'e git; diğer durumda, S_B ve E_B 'yi raporla ve bitir.

Çalışmada kolaylık olması açısından genetik algorithmada kullanılan kromozom gösterimi kullanılacaktır. Benzer gösterim bir başka çalışmada da kullanılmıştır (Alnahhal ve Noche, 2015). Şekil 2'de 12 istasyonlu bir montaj hattına hizmet veren 3 süpermarket gösterilmektedir. İlk süpermarket 1'den 5'e kadar olan istasyonlara hizmet verirken, ikinci süpermarket 6'dan 9'a kadar, üçüncüsü ise 10'da 12'ye kadar hizmet vermektedir. Süpermarketler istasyonların hepsine ardışık olarak hizmet verdiği düşünülürse, her süpermarketin hizmet verdiği son istasyonu, $\{5,9,12\}$ şeklinde ardışık olarak göstermek çalışmada kolaylık sağlayacaktır. Bu gösterimle hangi süpermarketlerin hangi istasyonlara hizmet verdiği kolaylıkla anlaşılacaktır.

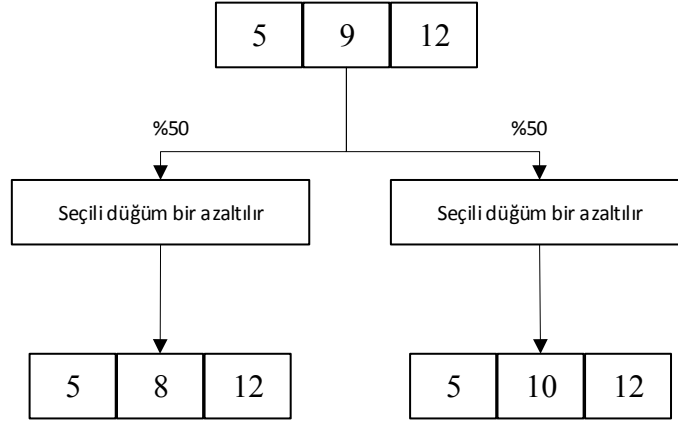


Çalışmadaki kromozom gösterimi

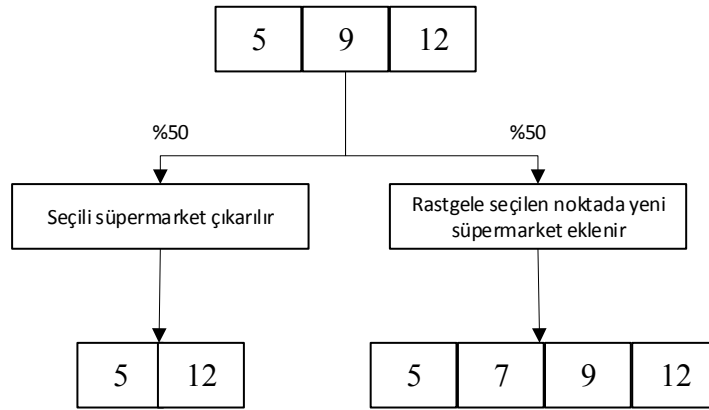
Şekil 2. Çalışmadaki kromozom gösterimi

Süpermarket Yerleşim Problemini TB algoritması ile çözmek için öncelikle rastgele bir başlangıç çözümü oluşturmak gerekir. Sonrasında komşu çözümler üretilerek, daha iyi çözümler araştırılır. Örneğin 12 istasyonu olan bir montaj hattı için 3 süpermarketli $\{5,9,12\}$ bir başlangıç çözümü olsun. Bu çözümde bir değer değiştirilerek $\{5,10,12\}$ şeklinde bir komşu çözüm üretilebilir. Çözüm iyileşmişse $\{6,10,12\}$ şeklinde yeni bir komşu çözüm üretilebilir. Çözümdeki değerleri değiştirmek, süpermarket sayısı aynı kalarak hizmet verilen istasyonları değiştirmek demektir. Fakat farklı sayıda süpermarket içeren çözümler de daha iyi sonuç verebileceği için komşu çözümler üretilirken süpermarket sayısının da değiştirilmesi gerekir.

Bu çalışmada TB algoritmasında komşu çözüm üretilirken iki yöntem kullanılmıştır. Birincisi geçerli çözümde rastgele bir değer seçilerek %50 olasılıkla bu değer bir azaltılmış ya da artırılmıştır. İkinci olarak da %50 olasılıkla geçerli çözümden rastgele seçilmiş bir süpermarket çıkarılarak süpermarket sayısı bir azaltılmış ya da rastgele seçilmiş bir noktada çözüme yeni bir süpermarket eklemiştir. Bu iki yöntem de %50 olasılıkla seçilmiştir. Şekil 3 ve Şekil 4 bu iki yöntemi {5,9,12} örnek çözümü üzerinde göstermektedir.



Şekil 3. Rastgele seçilen bir düğümün değerinin değiştirilmesi



Şekil 4. Rastgele seçilen bir süpermarketin çıkarılması ve rastgele seçilen noktada yeni bir süpermarket eklenmesi

Çalışmada önerilen komşu çözüm algoritması aşağıdaki gibidir:

Prosedür: Komşu çözüm algoritması

1. p1 isminde [0, 1] arasında rastgele bir değer üret.
2. Eğer $p1 \leq 0.5$ ise Adım 5'e git.
3. p2 isminde [0, 1] arasında rastgele bir değer üret.
4. Eğer $p2 > 0.5$ ise çözümden rastgele seçilmiş bir elemanı çıkar; diğer durumda çözüme rastgele yeni bir elman ekle ve Adım 7'ye git.

5. p_3 isminde $[0, 1]$ arasında rastgele bir deęer üret.
6. Eęer $p_3 > 0.5$ ise çözümden rastgele seçilmiş bir elemanı bir artır, dięer durumda bir azalt.
7. Bitir

3. Bulgular ve Tartışma

Çalışmada önerilen algoritmanın etkinlięi deęerlendirmek amacıyla, Alnahhal ve Noche (2015)'un çalışmasında önerilen yöntemle bir dizi test problemi oluşturulmuştur. İstasyonların talepleri 1 ile 10 arasında rastgele belirlenmiştir. İki istasyon arası mesafe x eksen koordinatlarına (a_s) 3 ile 6 arasında rastgele deęerler eklenerek elde edilmiştir. 20, 30, 40, 60, 100, 150 ve 200 olmak üzere farklı istasyon sayılarıyla (S) ve deęişen sabit maliyetlerle (r) 24 adet test problemi oluşturmuştur. Her bir test problemi için de istasyon koordinatları ve talepleri deęişen 4 farklı test problemi oluşturulmuştur.

Daha iyi çözümler elde edilebilmek için önerilen TB algoritmasının parametreleri ön deneylerle belirlenmiştir. T_0 , T_{min} , α parametrelerinin etkin deęerlerini bulabilmek için rastgele üç test problemi seçilmiştir. T_0 için üç seviye (500, 1000, 1500), T_{min} için üç seviye (“0,1”, “1”, “10”) ve α için üç seviye (“0,9”, “0,95”, “0,99”) belirlenmiş ve her parametre için her seviye üçer kez çalıştırılarak test edilmiştir. Elde edilen sonuçlar üzerinde, toplam maliyet ve CPU zamanı açısından varyans analizi yapılmış ve T_0 , T_{min} ve α parametreleri sırasıyla 1000, 1 ve 0,99 olarak belirlenmiştir.

Çalışmada önerilen algoritma kullanılarak MS Visual Studio ortamında C# programlama dili ile bir uygulama geliştirilmiş ve tüm test problemleri 32 GB bellek ve Intel Xeon 3.7 Ghz işlemcili bir masaüstü bilgisayar üzerinde çalıştırılmıştır. Her test problemi geliştirilen uygulama aracılığıyla beşer kez test edilmiş, çözümdeki ortalama süpermarket sayısı, ortalama maliyet ve ortalama CPU zamanı hesaplanmıştır.

Önerilen algoritmanın performansını kıyaslayabilmek için, birerleme yöntemiyle optimum sonuca ulaşan bir uygulama daha geliştirilmiştir. Tüm test problemleri bu uygulama aracılığıyla çalıştırılmış ve 7200 saniyelik çalışma sonrasında sonuç alınmadığı durumda uygulama sonlandırılmıştır. Uygulama küçük ölçekli (20 ve 30 istasyon içeren) test problemlerinde optimum çözüme ulaşabilmiş, büyük ölçekli (40 ve daha fazla istasyon içeren) test problemlerinde ise belirlenen sürede sonuç üretememiştir.

Tablo 3’de her bir test problemi için, istasyon sayısı ve sabit maliyet aynı kalmak şartıyla geliştirilen 4 farklı veri setinin önerilen algoritma ile beşer kez çalıştırılması sonucu elde edilen ortalama süpermarket sayısı, ortalama maliyet, ortalama CPU zamanı ve birerleme yöntemi ile hesaplanan optimum süpermarket sayısı ve optimum maliyetlerin ortalaması gösterilmiştir. Tablo 2’de, Tablo 3’de kullanılan kısaltmalar verilmiştir.

Tablo 3’de görülebileceği gibi geliştirilen uygulama küçük ölçekli tüm test problemleri ve veri setleri ile her çalıştırıldığında optimum sonuçlara eşit sonuçlar üretmiştir. Büyük ölçekli test problemlerinde ise birerleme yöntemi ile optimum sonuçlar hesaplamadığı için kıyaslama yapılamamıştır.

Tablo 2. Kısaltmaların gösterimi

S	İstasyon sayısı
r	Her süpermarket için sabit maliyet
Ort. n	Ortalama süpermarket sayısı
Ort. M.	Ortalama maliyet
T	Önerilen algoritma CPU zamanı
Opt. n	Ortalama optimum süpermarket sayısı
Opt. M.	Ortalama optimum maliyet
BT	Birerleme uygulaması CPU zamanı

Tablo 3. Uygulama sonuçları

No	S	r	Ort. n	Ort. M.	T (s)	Opt. n	Opt. M.	BT(s)
1	20	500	5,75	4917	0,1	5,75	4917	0,4
2		1000	4	6955,5	0,1	4	6955,5	0,4
3		1500	3,5	9553,5	0,1	3,5	9553,5	0,4
4		3000	2,25	13079	0,1	2,25	13079	0,4
5	30	500	9	7476	0,2	9	7476	534,8
6		1000	6,25	10865	0,1	6,25	10865	530,8
7		1500	5,25	14293	0,1	5,25	14293	527,8
8		3000	4	20495	0,1	4	20495	527,5
9	40	500	11,5	9661	0,2			
10		1000	8,5	14512,5	0,2			
11		1500	7	18877	0,1			
12		3000	5	28538	0,1			
13	60	500	18,55	15016,1	0,3			
14		1000	12,6	21939	0,2			
15		1500	10,35	27627,8	0,2			
16		3000	7,5	41589,8	0,1			
17	100	500	28,65	24531,5	0,5			
18		1000	21,3	37341,6	0,4			
19		5000	9,4	90738,4	0,2			
20		10000	6,8	128725,6	0,2			
21	150	1000	30,45	55138,7	0,5			
22		3000	18,25	102473,1	0,3			
23		5000	13,85	133079,8	0,3			
24		10000	10,05	195851,5	0,2			
25	200	1000	41,9	74736,3	0,7			

26		3000	24,35	137962,9	0,5			
27		5000	18,8	177663,9	0,4			
28		10000	13,25	258112,7	0,3			

4. Sonuçlar ve Öneriler

Bu çalışmada işletme zeminine kurulacak süpermarketlerin yerleşim problemi için Emde ve Boysen'in (2012) çalışmasında önerilen matematiksel modelden faydalanılarak bir tavlama benzetimi algoritması önerilmiştir. Önerilen algoritma kullanılarak bir uygulama geliştirilmiş ve algoritmanın başarısı bir dizi test problemi ile ölçülmüştür. Birerleme yöntemiyle bir uygulama daha geliştirilmiş, test problemlerinin optimum sonuçları hesaplanmış ve önerilen algoritmanın test sonuçları ile karşılaştırılmıştır. Küçük ölçekli test problemleri için önerilen algoritma, yaklaşık saniyenin onda biri çalışma sürelerinde optimum sonuçlara eşit sonuçlar üretmiştir. Büyük ölçekli test problemleri için ise, birerleme yöntemi ile sonuç bulunamadığı için karşılaştırma yapılamamıştır. Buna rağmen küçük test problemlerinde her çalıştırmada optimum sonuçlara ulaşıldığı için, büyük test problemlerinde de algoritmanın başarılı olarak çalışacağı söylenebilir. Birerleme yönteminde, 20 istasyonlu test problemlerinden 30 istasyon içeren test problemlerine geçildiğinde, çalışma zamanının yaklaşık 1300 kat arttığı gözlenmiştir. Bundan yola çıkarak 40 ve daha fazla istasyon içeren test problemlerin birerleme yöntemi ile sonuca ulaşmanın günler süreceği söylenebilir. Önerilen tavlama benzetimi algoritması büyük ölçekli test problemlerinde de bir saniyenin altında sonuçlar üretebilmiştir.

Literatürde SYP ile ilgili çok az sayıda çalışma vardır. Bunların içinde sadece birkaç tanesi probleme çözüm önerisi getirmiştir. Problemin çözümü için dinamik programlama ve genetik algoritma gibi yöntemler kullanılmıştır. Bu çalışmada komşuluk yapısı içeren algoritmaların da problem çözümünde etkili olabileceği düşünülerek tavlama benzetimi algoritması kullanılmıştır. Sonuçta geliştirilen algoritmanın düşük çalışma sürelerinde başarılı sonuçlar elde edildiği görülmüştür.

Kaynaklar

- Alnahhal, M., and Noche, B. (2015). A genetic algorithm for supermarket location problem. *Assembly Automation*, 35(1), 122-127.
- Battini, D., Gamberi, M., Persona, A., and Sgarbossa, F. (2015). Part-feeding with supermarket in assembly systems: transportation mode selection model and multi-scenario analysis. *Assembly Automation*, 35(1), 149-159.
- Battini, Daria, Boysen, N., and Emde, S. (2013). Just-in-Time supermarkets for part supply in the automobile industry. *Journal of Management Control*, 24(2), 209-217.
- Battini, Daria, Faccio, M., Persona, A., and Sgarbossa, F. (2010). "Supermarket warehouses": stocking policies optimization in an assembly-to-order environment. *The International Journal of Advanced Manufacturing Technology*, 50(5-8), 775-788.

- Boysen, N., Emde, S., Hoeck, M., and Kauderer, M. (2015). Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research*, 242(1), 107-120.
- Emde, S., and Boysen, N. (2012). Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *International Journal of Production Economics*, 135(1), 393-402.
- Faccio, M., Gamberi, M., and Persona, A. (2013). Kanban number optimisation in a supermarket warehouse feeding a mixed-model assembly system. *International Journal of Production Research*, 51(10), 2997-3017.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-679.
- Metropolis, N., Rosenbluth, A. W., and Teller, A. H. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 1087-1092.
- Nourmohammadi, A., Eskandari, H., Fathi, M., and Aghdasi, M. (2018). A mathematical model for supermarket location problem with stochastic station demands. *Procedia CIRP*, 72, 444-449.
- Özcan, U., and Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. *Computers and Industrial Engineering*, 57(1), 217-227.
- Wanke, P. F., and Zinn, W. (2004). Strategic logistics decision making. *International Journal of Physical Distribution and Logistics Management*, 34(6), 466-478.
- Yano, C. A., Bozer, Y., and Kamoun, M. (1998). Optimizing dock configuration and staffing in decentralized receiving. *IIE Transactions*, 30(7), 657-668.