

# GENEL AMAÇLI ARAMA ALGORİTMALARI İLE BENZETİM ENİYİLEMESİ: EN İYİ KANBAN SAYISININ BULUNMASI

Hüseyin GÜDEN, Barış VAKVAK, Barış E. ÖZKAN, Fulya ALTIPARMAK, Berna DENGİZ  
*Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü*

## ÖZET

Gerçek hayatta karşılaşılan sistemler büyük boyutlu ve karmaşık olup birtakım belirsizlikler içerdiğinde analitik çözümleri mümkün olmamaktadır. Bu tür sistemlerin bu karmaşık ve olasılıklı yapısından dolayı benzetim, mevcut sistemlerin performans ölçütlerinin tahmini için analiz ve yeni kurulacak sistemler için bir tasarım aracı olarak kullanılmaktadır. Benzetim modeli, bir girdi setine karşılık bir çıktı setinin elde edildiği girdi-çıkıtı modelidir. Benzetim tek başına bir eniyileme tekniği değildir. Benzetim en iyi performansı sağlayacak karar değişkenleri hakkında tahmini değerler elde edilmesini sağlar. Ancak, çözüm uzayı büyüdükçe, karar değişkenlerinin oluşturacağı tüm kombinasyonların denenmesi benzetim koşum maliyetini katlanılmaz hale getirmektedir. Bu nedenle son yıllarda yaygın kullanılan bir yaklaşım, benzetim modeli ve arama algoritmalarının birlikte kullanıldığı benzetim eniyilemesidir. Bu çalışmada, gerçek cep telefonu üretiminden esinlenerek oluşturulmuş düşünsel bir tam zamanında üretim sistemi için, en iyi kanban kombinasyonunun bulunması amacıyla, genel amaçlı sezgisel teknikler sınıfında yer alan Tabu Arama (TA), Genetik Algoritma (GA) ve Tavlama Benzetimi (TB) algoritmaları sistemin benzetim modeli ile birlikte kullanılmıştır. Algoritmaların performansı rassal arama tekniği de dikkate alınarak çözüm zamanı ve çözüm kalitesi açısından karşılaştırılmıştır.

**Anahtar Kelimeler:** Kanban, tam zamanında üretim sistemi, benzetim eniyilemesi, genetik algoritmalar, tavlama benzetimi, tabu arama, rassal arama.

## ABSTRACT

When the systems under investigation are complex and inherit uncertainties, the analytical solutions to these systems become impossible. Because of the complex stochastic characteristics of such systems, simulation can be used as an analysis tool to predict the performance of an existing system or a design tool to test new systems under varying circumstances. Simulation model in which the set of output is estimated for given a particular set of input is an input-output model. Since simulation is not an optimization tool, the difficulty is then to know how to drive simulation experiments in order to determine the value of decision variables. In fact, when there is large set of decision variables with many possible values, the number of possible combinations is such that an exhaustive search is not possible. Recently, search techniques have been used with simulation model to overcome this difficulty. In this study, Tabu Search (TS), Genetic Algorithms (GA) and Simulated Annealing (SA) belong to general purpose search techniques are applied to find optimum combination of kanbans on a hypothetic Just in Time production system stemmed from a real production system of mobile phone. The effectiveness and efficiency of the general purpose search algorithms together with random search are investigated according to solution quality and solution time.

**Key Words:** Kanban, Just in time production system, simulation optimization, genetic algorithms, simulated annealing, tabu search, random search.

## GİRİŞ

Gerçek hayatta karşılaşılan birçok sistem, genellikle büyük boyutlu, karmaşık ve birtakım belirsizlikler içermektedir. Bu tür sistemlerdeki problemlerin çözümü bilinen eniyileme metotlarıyla çok zor kimi zaman da imkansız olmaktadır. Bu tür sistemlerin analizi amacıyla çoğunlukla benzetim kullanılmaktadır. Benzetim modeli, bir girdi setine karşılık bir çıktı ya da çıktı setinin elde edildiği girdi-çıkıtı modelidir. Benzetim modeli; sistemin davranışını, analizini, ilgilenilen performans ölçütlerinin tahminini, farklı sistem parametrelerinin sonuç üzerindeki etkisini gözleme imkanı veren bir araçtır.

Eniyileme, problemin karar değişkenlerinin mümkün tüm kombinasyonları arasından en iyi performansı (en iyi amaç fonksiyonu değerini) veren kombinasyonun bulunmasıdır. Bu amaçla literatürde birçok eniyileme algoritması önerilmiştir. Bu algoritmaların çoğu sistemin modeli ve amaç fonksiyonu için matematiksel modellere ihtiyaç duymaktadır. Karmaşık sistemler için matematiksel modelin kurulması çoğu zaman zordur. Model kurulsa bile çözüm zamanı maliyeti çok yüksek olduğundan kullanılamamaktadır. Dolayısıyla, bu tür karmaşık sistemlerin eniyilenmesi amacıyla benzetim modelinin kullanılması kaçınılmaz olmaktadır.

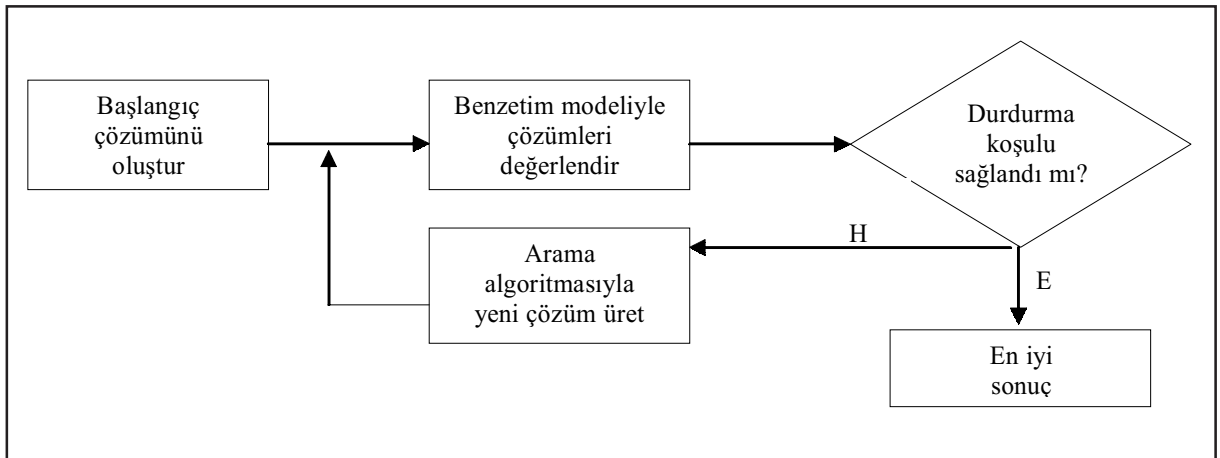
Benzetim tek başına bir eniyileme tekniği olmayıp sadece karar değişkenlerinin bir kombinasyonu için

tahmini değerlerin elde edilmesini sağlar. Dolayısıyla, çözüm uzayı büyüdükçe karar değişkenlerinin oluşturacağı tüm kombinasyonların benzetim ile denemesi imkansızdır. Bu nedenle, benzetim modeli ve eniyileme algoritmalarının birlikte kullanıldığı benzetim eniyilemesi teknikleri günümüzde yaygın bir şekilde kullanılmaktadır.

Arama algoritmaları, algoritmanın yapısına uygun olarak seçilen bir başlangıç çözümünden ya da çözümlerinden başlar ve farklı yapılar kullanarak yeni çözümlere ulaşırlar. Bu çözümler amaç fonksiyonu (benzetim modeli) ile değerlendirilir. Bu akış belirli bir durdurma koşulu sağlanıncaya kadar devam eder ve bu koşula ulaşıncaya kadar devam eder ve bu koşula ulaşıncaya kadar devam eder ve bu koşula ulaşıncaya kadar devam eder. Şekil 1'de benzetim eniyilemesi amacıyla kullanılan arama algoritmalarının çalışmasını açıklayan akış şeması görülmektedir.

Bu çalışmada, gerçek cep telefonu üretiminden esinlenerek oluşturulmuş düşünsel bir tam zamanında üretim (TZÜ) sistemi için eniyi kanban kombinasyonunun bulunması amacıyla genel amaçlı sezgisel teknikler sınıfında yer alan Tabu Arama (TA), Genetik Algoritma (GA) ve Tavlama Benzetimi (TB) algoritmaları kullanılmıştır. Bu algoritmaların benzetim eniyilemesindeki performansları çözüm kalitesi ve çözüm zamanı açısından karşılaştırılmıştır.

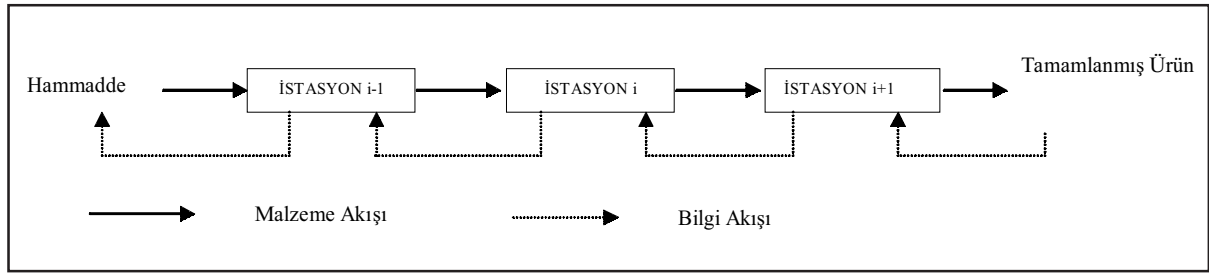
Bilindiği gibi TZÜ sistemi bir çekme sistemidir. Çekme sisteminin temel amacı; istasyonlar arası stoku, boş



Şekil 1. Benzetim Eniyilemesi Algoritmaları Akış Şeması

zamanları, hataları ve dalgalanmaları enküçükmektir. İdeal bir çekme sisteminde her istasyondaki stok miktarı sıfır olmalıdır. Bu tür sistemlerde çıktı düzeyi itme sistemlerine kıyasla, işlem zamanlarındaki değişim oranlarına daha duyarlıdır ve iyi bir çıktı düzeyi elde etmek için aşamalar arasındaki işlem zamanlarının değişimini düzenlemek gerekir. Yarı işlenmiş ürünün öncül istasyondan ardıl istasyona, ardıl istasyon talep ettikçe gönderildiği çekme sisteminde malzeme isteğinin yapılması ve üretimin gerçekleştirilmesi için kanban sistemi kullanılır. Şekil 2'de bu tür sistemlerde malzeme ve bilgi akışı görülmektedir. Kanban sistemi, parçaların üretimini kontrol eden bir bilgi sistemi olup ana özelliği

yapılmıştır. Huang vd. (1983), çok hatlı, çok aşamalı TZÜ sistemlerinde değişken işlem zamanlarının, darboğazların, değişken talebin ve değişik kanban sayılarının sistemin performansı üzerindeki etkilerini araştırmışlardır. Gupta ve Gupta (1989), dinamik koşullarda konteynır ve kanban sayılarının etkisini TZÜ sisteminin benzetimini yaparak incelemişlerdir. Rees vd. (1987), TZÜ sistemindeki kanban sayılarını düzenlemek üzere kullanılan sipariş varış zamanını hesaplayan dinamik bir prosedür geliştirmişlerdir. Wang ve Wang (1991), ardışık iki istasyon arasındaki eniyi kanban sayısını bulmak için Markov sürecini kullanmışlardır. Fukukawa ve Hong (1993), çok hatlı, çok aşamalı TZÜ



Şekil 2. Çekme Sisteminin Malzeme ve Bilgi Akışı

iş istasyonunda üretimle ilgili karar vermeyi sağlayacak bilginin hazır olmasıdır. Ayrıca, varsa talep değişiminin iletişimini sağlar ve ardıl aşamanın acil ihtiyaçlarını zamanında temin eder. TZÜ sisteminde bu bilgiler kanban adı verilen kartlar ile iletilir. Bu kartlar konteynırlarla veya taşıyıcılarla iletilir. TZÜ sisteminde farklı tipte kanban söz konusu olmasına rağmen en çok kullanılan iki kanban tipi çekme ve üretim kanbanıdır. Çekme kanbanı, ardıl istasyon için öncül istasyondan yarı işlenmiş ürün isterken üretim kanbanı bir iş istasyonunda üretimin başlamasını sağlar (Nahmias, 1993).

Bir TZÜ üretim sisteminin performansı istasyonlar arasına yerleştirilecek kanbanların sayısına çok bağlıdır. Bu nedenle literatürde dinamik koşullarda istasyonlar arasında kanban sayısının belirlenmesi veya farklı kanban sayılarının sistemin performansı üzerindeki etkisinin incelenmesi üzerine çeşitli çalışmalar

sistemleri için dinamik koşullarda istasyonlar arasındaki eniyi kanban sayılarını belirlemek için bir tamsayı amaç programlama modeli geliştirmişlerdir.

Bu çalışmada dikkate alınan dört hatlı ve tek ürünli düşünsel TZÜ sisteminde çekme ve üretim kanbanları kullanılmaktadır. Amaç, yarı ürünleri stokta tutma maliyeti ile müşterileri bekletmenin (talebin zamanında karşılanamaması) maliyetinden oluşan toplam maliyeti en azlayacak istasyonlar arasındaki çekme ve üretim kanban sayılarını belirlemektir.

Çalışmanın 2. Bölümünde sistem ve problemin tanımı verilmektedir. 3. Bölümde genel amaçlı arama algoritmaları açıklanarak benzetim eniyilemesinde bu algoritmaların kullanımı ile ilgili literatür araştırması verilirken, 4. Bölümde bu çalışmada kullanılan arama algoritmaları detaylı olarak incelenmektedir. Son bölümde ise algoritmalar, çözüm kalitesi ve çözüm zamanı açısından karşılaştırılmaktadır.



**Tablo 1.** İstasyonlara Ait İşlem Zamanları ve Dağılımları

	İşin Adı	Dağılım Türü	Parametre (dakika)
1. hat	Makina dizgi	Sabit	4
	Elde lehim	Üçgen	(4,6,8)
	Göz kontrol	Üçgen	(9,10,11)
	Test	Üçgen	(4,5,6)
2. hat	Makina dizgi	Sabit	1
	Göz kontrol	Üçgen	(4.7,5,5.4)
	Test	Üçgen	(1.6,2,2.5)
3. hat	Makina dizgi	Sabit	1
	ICCT test	Sabit	1
4. hat	Montaj (pil)	Üçgen	(4.5,5,5.5)
	Ultrasonik kaynak	Sabit	0.1
5. hat	Cihaz montaj	Üçgen	(6.5,7,7.5)
	Son test	Üçgen	(9,10,11)
	Tamir	Üstel	2
	Paketleme	Uniform	(0.8,1.5)

Bu varsayımlar altında sistemin benzetim modeli SIMAN benzetim diliyle, GA, TA ve TB algoritmaları ise C diliyle kodlanmış ve bütünleşik olarak UNIX ortamında çalıştırılmıştır.

### GENEL AMAÇLI ARAMA ALGORİTMALARI

Bu bölümde, bu çalışmada dikkate alınan TZÜ sisteminde yeni kanban kombinasyonunun bulunması için kullanılan genel amaçlı arama algoritmaları olan tavlama benzetimi, genetik algoritmalar ve tabu arama detaylı olarak incelenecek ve bu algoritmaların benzetim eniyilemesi amacıyla kullanımına ilişkin literatür araştırması verilecektir.

#### Tavlama Benzetimi Algoritması

TB algoritması, katıların belirli bir başlangıç sıcaklığından başlayarak yavaş yavaş soğutulduğu tavlama sürecinin benzetimi olan stokastik bir arama algoritmasıdır. Kirkpatrick, Gerlatt ve Vecchi (1983) ve Cerny (1985) tarafından ayrı ayrı önerilmiştir. Son yıllarda birçok araştırmacı TB algoritmasını kombinatoriyal eniyileme problemlerinde kullanmaktadır.

Tavlama terimi fiziksel olarak, ısı banyosundaki bir katının yüksek enerji durumlarından başlayarak daha düşük enerji durumlarının elde edilme sürecini temsil etmektedir. Bu süreç çok genel olarak iki işlemden oluşmaktadır:

- Isı banyosunun başlangıç sıcaklığının katının eriyebileceği bir değere kadar yükseltilmesi.
- Katılar düşük enerjili durumda, yani düşük sıcaklıkta daha kararlıdır. Yani katıların parçacıkları düşük sıcaklıklarda daha düzenlidir. Bundan dolayı katının parçacıkları kendini düzenleyene kadar ısı banyosunun sıcaklığının giderek azaltılması.

Katının sıvı durumunda tüm parçacıkları gelişi güzel hareket ederler. Katı durumda ise bir kafes şeklinde düzenlenirler. Bu durumda sistemin enerjisi en azdır ve bu duruma yer durumu denmektedir. Bir katının yer durumu, sıcaklık yeteri kadar yükseltilmiş ve soğutma da yeteri kadar yavaş yapılmışsa elde edilir. Aksi halde katının bulunduğu durum yarı kararlı bir durumdur (Alabaş, 1999).

Eniyileme problemleriyle TB arasındaki benzerlikler aşağıdaki gibi gösterilebilir:

- Katının farklı fiziksel durumları problemdeki mümkün çözümlere,
- Sistemin enerjisi amaç fonksiyonuna,
- Bir durumun enerjisi bir çözümün amaç fonksiyonu değerine,
- Yarı kararlı durum yerel eniyi çözüme,
- Yer durumu genel eniyi çözüme karşılık gelir.

Katıların bu tavlama sürecinde geçirdiği durumları benzetebilmek için Metropolis vd (1953) tarafından "Metropolis Algoritması" geliştirilmiştir. Algoritmada i durumunda bulunan katının enerjisi  $E_i$  iken, bir sonraki j durumuna geçen katının enerjisi  $E_j$ 'dir. Eğer j durumundaki enerji i durumundaki enerjiden küçük veya eşitse, j durumu yeni mevcut çözüm olarak kabul edilir. Aksi takdirde j durumu 1 nolu formüle göre elde edilen olasılık değeri ile kabul edilir. Bu olasılık değeri "Metropolis Kriteri" olarak anılır. Burada  $k_B$  fiziksel tavlama sürecinde "Boltzman Sabiti" olarak bilinen fiziksel bir sabiti ifade eder.

$$\exp\left[\frac{E_i - E_j}{k_B * T}\right] \quad (1)$$

TB algoritmasının amacı, tüm mümkün çözüm noktalarının bir alt kümesinde (S) tanımlanmış bir  $f(x)$  fonksiyonunu eniyileyecek bir x çözümü bulmaktır. TB algoritması rassal olarak seçilen bir başlangıç çözümüyle aramaya başlar. Bundan sonra uygun bir mekanizma ile bu çözüme komşu bir çözüm seçilir ve  $f(x)$ 'de meydana gelen değişim hesaplanır. Eğer değişim istenilen yönde ise komşu çözüm mevcut çözüm olarak alınır. Eğer istenen yönde bir değişim sağlanmamışsa, TB algoritması bu çözümü "Metropolis Kriteri" ile elde edilen olasılık değeri ile kabul eder. Amaç fonksiyonunda ters yönde bir değişim yaratan bir çözümün belli olasılık değeri ile kabulü, TB algoritmasının yerel eniyi noktalardan kurtulmasını sağlamaktadır. Yukarıdaki olasılık değerine göre T değeri yüksek olduğunda amaç fonksiyonunda meydana gelen artışların çoğu kabul edilecektir. T değeri azaldıkça kabul edilme oranı da azalacaktır. Bu nedenle TB

algoritmasında yerel noktalara takılmamak için başlangıç sıcaklık değeri yüksek seçilerek yavaş yavaş azaltma yoluna gidilmektedir.

### Genetik Algoritmalar

GA, biyolojik sistemlerin doğal evrim mekanizmasının benzetimini yapan stokastik bir arama algoritmasıdır (Goldberg,1989). GA ilk defa 1970'lerin başında John Holland tarafından ortaya atılmıştır. Literatürde bulunan eniyileme metotları ile çözülmesi çok zor, kimi zaman da imkansız olan ya da çözüm zamanı problemin büyüklüğü ile üstel olarak artan NP-zor problemlerde kullanılan GA, eniyi ya da eniyiye yakın çözüm vermektedirler. Doğada geçerli olan "eniyinin yaşaması" prensibine dayalı olarak çalışan GA sürekli iyileşen çözümler üretir. Önceleri sadece doğrusal olmayan eniyileme problemlerine uygulanan GA, daha sonra karesel atama, çizelgeleme, gezgin satıcı, tesis yerleşimi, montaj hattı dengeleme ve şebeke tasarımı gibi kombinatoriyal eniyileme problemlerine de başarıyla uygulanmıştır (Altıparmak, 1996).

GA, yığın olarak adlandırılan, mümkün çözümlerin kodlandığı dizilerle biyolojik sistemlerin özelliğini benzetilen operatörlerden oluşur. GA'nın ilk adımı başlangıç yığınının oluşturulmasıdır. Başlangıç yığını problemin tüm mümkün çözümlerinin bir alt kümesinden oluşmaktadır. Bu yığındaki her bir dizi bir kromozom olarak kodlanmaktadır. Yığın genişliğinin seçiminde problemin karmaşıklığı ve aramanın derinliği önemlidir. Bir yığında bulunan bireylerin ne kadar iyi olduğu "uygunluk fonksiyonu" ile değerlendirilir. Bu nedenle öncelikle dizilerin uygunluk değerleri hesaplanır. Basit bir GA' da probleme özgü çalışan tek kısım uygunluk fonksiyonudur. Uygunluk fonksiyonu, dizileri problemin parametreleri haline getirerek onların bir bakıma şifresini çözer ve bu parametreleri kullanarak dizilerin uygunluğunu bulur. Çoğu zaman GA'nın başarısı bu fonksiyonun verimli ve hassas olmasına bağlıdır. İyi bir dizi, enbüyükleme problemi için yüksek, enküçükleme problemi için düşük uygunluk değerine sahip olmalıdır (Michalewicz,1992).

GA'da bir yığından gelecek ve yeni yığını oluşturacak bireyler bir seçim mekanizması ile seçilirler. Mevcut yığından gelecek yığına geçecek olan bireyler mevcut yığında en uygun değere sahip olan bireylerdir. GA'da bu işleme yeniden üretim işlemi denir. Yeni yığın oluşturulduktan sonra bu yığındaki bireyler çaprazlama ve mutasyon operatörleri kullanılarak değişime uğratılır. Bu operatörler, yeniden üretim işlemi ile seçilen yığındaki bireylerin bir kısmına belirli olasılık değerleri ile uygulanarak mevcut yığından farklı noktaların elde edilmesini sağlarlar. Çaprazlama operatörü farklı iki birey arasında bilgi değişimini, mutasyon operatörü ise bir bireyde rassal değişimi sağlayarak yeni bireylerin oluşmasına olanak verir. Bu işlem sonunda edilen dizilerin uygunluk değerleri tekrar hesaplanır. Belirlenen durdurma koşulu sağlandığında yığındaki eniyi uygunluk değerine sahip dizi problemin çözümü olarak alınır (Dengiz ve Altıparmak, 1998).

Çaprazlama operatörü yığın içindeki farklı bireyler arasında gen değişiminin yapılarak, aynı yığın içinde farklı noktalara doğru aramanın gerçekleşmesini sağlar. Bir yığına çaprazlama operatörü ( $p_{cap}$ ) olasılığı ile uygulanır. Tek noktalı çaprazlama, 2 noktalı çaprazlama, uniform çaprazlama çok sık kullanılan çaprazlama operatörleridir (Goldberg, 1989). GA son iterasyonlarda iyi çözümlere yakınsadığından dolayı yığındaki bireyler birbirine çok benzemektedir. Bu aşamada sadece çaprazlama operatörünün kullanılması yığındaki değişkenliği sağlayamamaktadır. Mutasyon operatörü ise dizi içinden seçilen bir veya birkaç geni değiştirerek yığında değişkenliğin meydana gelmesine yardımcı olmaktadır. İkili düzende kodlamanın yapıldığı bir dizide rassal olarak seçilen bir veya birden fazla elemanın değeri 0 ise 1, 1 ise 0 yapılarak mutasyona uğratılır. Böylece yeni bir birey elde edilmiş olur. Mutasyonun etkinliği özellikle algoritmanın son adımlarında kendini göstermektedir. Her yığına mutasyon operatörü ( $p_{mut}$ ) olasılığı ile uygulanır.

### **Tabu Arama Algoritması**

TA algoritması, kombinatoriyal eniyileme problemlerini çözmek üzere geliştirilmiş ve tesis yerleşimi,

çizelgeleme, ulaştırma, parti büyüklüğü gibi birçok kombinatoriyal eniyileme problemlerine başarıyla uygulanmıştır (Alabaş, 1999). TA algoritmasının bugünkü şekli, Glover tarafından 1989 yılında ortaya konulmuştur (Banks ve diğerleri, 2001). TA algoritmasının en temel elemanları ve bunların işleyiş şekli aşağıda sırasıyla verilmektedir.

**Başlangıç Çözümünün Oluşturulması:** En genel şekilde başlangıç çözümü rassal olarak elde edilir. Ancak, ilgilenilen problem için geliştirilmiş olan bir sezgisel algoritmadan yararlanarak da başlangıç çözümünün elde edilmesi mümkündür.

**Hareket Mekanizması:** Mevcut bir çözümde yapılan bir değişiklik yeni çözümün elde edilmesi hareket mekanizmasıyla gerçekleştirilir. Hareket mekanizmasındaki olası hareketler, mevcut çözümün komşularını oluşturur. Hareket mekanizması algoritmanın etkinliği açısından önemli olduğu için problemin yapısına bağlı olarak uygun bir şekilde seçilmelidir.

**Aday Liste Stratejileri:** TA algoritması yapılması mümkün olan, tabu olmayan ve amaç fonksiyonunun değeri açısından eniyi sonucu veren hareketlerin seçilmesi kuralına dayalı olarak çalışır. Aday liste stratejileri mümkün hareket listeleridir. Bu listelerden hareketler belirli stratejilere göre seçilir.

**Hafıza:** TA algoritmasının temel elemanlarından birisi de hafızadır. Arama boyunca ortaya çıkan durumlar, H hafızasına kaydedilir. Bu hafıza kısa dönemli hafıza olarak adlandırılır. Yapılmasına izin verilmeyen hareketler "tabu" olarak adlandırılır ve esnek hafıza içinde "tabu listesi" adı altında kaydedilirler. Bu hareketler belli bir süre sonra tabu listesinden çıkarılır ve yapılmasına izin verilir.

**Tabu Yıkma Kriterleri:** Tabu yıkma kriterleri, tabunun ortadan kalkabileceği durumları ifade etmektedir. En genel tabu yıkma kriteri, mevcut durumdan daha iyi bir sonuç verecek tabu hareketinin yapılmasına izin verilmesidir. Bu kriterin kullanılması TA algoritmasının etkinliğini artırmaktadır. Ayrıca, eğer tüm mümkün hareketler tabu ise bu hareketlerden tabu süresinin bitmesine en yakın olan bir tabu hareketine izin verilir.

**Durdurma Koşulu:** TA algoritması, bir veya birden fazla durdurma koşulunu sağlayıncaya kadar aramasını sürdürmektedir. Bu koşullardan bazıları aşağıda verilmiştir.

- Seçilen bir komşu çözümün komşusunun olmaması
- Belirli bir iterasyon sayısına ulaşılması
- Belirli bir çözüm değerine ulaşılması
- Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi

TA algoritması, bir başlangıç çözümü ile aramaya başlar. Algoritmanın her iterasyonunda tabu olmayan bir hareket ile mevcut çözümün komşuları içerisinde bir tanesi seçilerek değerlendirilir. Eğer amaç fonksiyonunun değerinde bir iyileştirme sağlanmışsa komşu çözüm, mevcut çözüm olarak dikkate alınır. Seçilen bir hareket tabu olmasına rağmen tabu yıkma kriterlerini sağlıyorsa, mevcut çözümü oluşturmak için uygulanabilir. Geriye dönüşleri önlemek için, bir takım hareketler tabu listesine kaydedilerek tekrar yapılması belirli bir süre için yasaklanır. Belirlenen bir durdurma koşuluna göre algoritmanın çalışması sonlanmaktadır.

### Literatür Araştırması

Bulgak ve Sanders (1988), eşzamansız montaj sisteminde ara stok seviyelerinin belirlenmesinde TB algoritmasını kullanmışlardır. Haddock ve Mittenthal (1992), düşünsel bir üretim sisteminde toplam beklenen kâr en büyük yapacak parametre düzeylerini bulmak için TB önermişlerdir. Bu çalışmada, uygun tavlama planının mevcut olmadığı durumlarda da TB algoritmasının  $n$  kez ( $n > 1$ ) çalıştırılarak eniyi ya da eniyeye yakın çözümü bulabildiğini göstermişlerdir. Yunker ve Tew (1994), bir üniversitedeki bilgisayar sisteminin eniyilenmesi amacıyla GA, örnek (pattern) arama ve cevap yüzeyi metotlarını karşılaştırarak eniyi maliyetin GA ile elde edildiğini ancak GA'nın çalışma zamanının diğer iki metoda göre daha fazla olduğunu göstermişlerdir. Pierreval ve Tautou (1997), GA' yı plastik yoğurt kapları üreten bir üretim sisteminin eniyilenmesi için kullanmışlardır. Bu çalışmada deterministik ve stokastik olmak üzere iki farklı sistem dikkate alınmıştır. Ahmed ve arkadaşları (1997),

basitten karmaşığa doğru oluşturdukları üç farklı düşünsel sistemde, sistemin çalışma maliyetini enküçüklemeye çalışmışlardır. Paul ve Chaney (1998), çelik ürünleri üreten bir fabrikada benzetim eniyilemesi amacıyla GA' yı kullanmışlardır. Çalışmada, GA ile atık demir (zayıat) miktarı sıfırlanırken çıktı oranı artırılmıştır. Dengiz ve arkadaşları (1998), periyodik gözden geçirmeli stok sisteminde eniyi ( $s, S$ ) politikasını ve eşzamansız otomatik montaj sisteminde eniyi ara stok seviyelerinin belirlenmesi amacıyla GA' yı kullanmışlardır. Ahmed ve Alkhamis (1998), kesikli stokastik eniyileme problemlerini çözmek için TB ile sıralama ve seçim metotlarının birlikte kullanıldığı yeni bir yaklaşım önermişlerdir. Önerilen algoritmada her seviye için belli sayıda arama yapılarak genel eniyi aranmaktadır. Bu algoritmada, TB algoritmasının her iterasyonunda sıralama ve seçim algoritması kullanılarak yeni komşular elde edilmektedir. Azadivar ve Tompkins (1999), kalitatif değişkenlerin de söz konusu olduğu bir sistemde benzetim eniyilemesi amacıyla GA' yı kullanmışlardır. Çalışmada, kalitatif karar değişkenlerine sahip sistemler için nesne odaklı benzetim modelleriyle GA' ları birlikte kullanmışlardır. Ayrıca, dikkate alınan problemler için GA ve rassal arama algoritmaları çözüm kalitesi açısından karşılaştırılmış ve GA' nın rassal aramaya göre daha iyi olduğu gösterilmiştir. Pierreval ve Paris (2000), 5 makinalı, tek ürünlü akış tipi üretim sisteminde "Bölünmüş Evrim Algoritmaları Yöntemi" ni kullanmışlardır. Kendi içinde arama yapan ve birbirleriyle etkileşimli çalışan bu algoritmalara "Yerel Evrim Algoritmaları (YEA)" adını vermişlerdir. Çalışmada, YEA ile büyük arama uzayının daha kısa zamanda incelenebildiği gösterilmiştir. Pierreval ve Paris (2001), 4 makinalı, 3 ürünlü tam zamanında üretim sisteminde taşıyıcı sayısı, ara stoklar ve ertelenen siparişleri dikkate alarak maliyetin enküçüklenmesi probleminde yine "Bölünmüş Evrim Algoritmaları" yaklaşımını kullanarak benzer sonuçlara ulaşmışlardır. Alabaş ve arkadaşları (2002), benzetim eniyilemesi amacıyla GA, TB, TA ve yapay sinir ağlarının performansını bir hatlı çok ürünlü çok aşamalı bir TZÜ sistemi üzerinde incelemiştir.



Bu çalışmada ise dört paralel hattan gelen dört ayrı yarı ürünün montajının yapıldığı karmaşık gerçek bir cep telefonu üretim sistemi TZÜ sistemi felsefesi ile modellenmiş ve eniyi kanban sayısının bulunmasında arama algoritmalarının performansı incelenmiştir.

## GENEL AMAÇLI ARAMA ALGORİTMALARI İLE ENİYİ KANBAN SAYISININ BULUNMASI

Genel amaçlı arama algoritmalarının TZÜ sisteminde eniyi kanban sayısının bulunmasına uygulaması ve parametrelerinin belirlenmesi ile ilgili detaylı bilgi bu bölümde verilecektir. Bu açıklamalardan önce algoritmalarda kullanılan parametreler aşağıda tanımlanmaktadır.

- $T$  : Fiziksel tavlamadaki sıcaklığa karşılık gelen kontrol parametresi  
 $f(x)$  : Maliyet fonksiyonu  
 $x_0$  : Başlangıç çözümü  
 $x_y$  : Kabul edilen yeni çözüm  
 $T_0$  : Başlangıç sıcaklığı  
 $T_s$  : Bitiş sıcaklığı  
 $I$  : Farklı noktalar için hesaplanan amaç fonksiyonu değerlerinin farkı  
 $x_a$  :  $x_y$ 'ye komşu olan aday çözüm  
 $r$  : Sıcaklık azaltma katsayısı  
 $k$  : Her sıcaklık değerinde aranan çözüm sayısı  
 $k_s$  : Her sıcaklık değerinde aranacak toplam çözüm sayısı  
 $m$  : İterasyon sayacı  
 $x^*$  : Eniyi çözüm  
 $x_k$  : Komşu çözüm  
 $x_{ek}$  : Eniyi komşu çözüm  
 $n$  : Yığındaki dizi sayısı  
 $p_{cap}$  : Çaprazlama oranı

$p_{mut}$  : Mutasyon oranı

$x_{meç}$  : Mevcut yığındaki eniyi çözüm

**Çözüm Kodlaması:** Bu çalışmada incelenen TZÜ sisteminde, 14 üretim 13 çekme kanbanı olmak üzere toplam 27 kanban vardır. Dolayısıyla arama algoritmaları için bir çözüm, kanban sayılarını gösteren tamsayıların bir dizisinden oluşmaktadır. Her dizi iki bölümden oluşmaktadır. Birinci bölüm her istasyondaki üretim kanbanı sayısını gösterirken ikinci bölüm ilgili istasyonlar arasındaki çekme kanbanı sayısını göstermektedir. Şekil 4'de bir çözümü gösteren bir dizi örnek olarak verilmektedir. Örnek dizi incelendiğinde 1. istasyonda iki üretim kanbanı varken 2. istasyondaki kanban sayısının üç olduğu görülmektedir. Aynı şekilde 1 ve 5 nolu istasyonlar arasındaki çekme kanbanı sayısı 2 iken, 2 ile 6 nolu istasyonlar arasındaki kanban sayısının ise 6 olduğu görülmektedir. Üretim ve çekme kanban sayısının alt ve üst sınırının belirlenmesi için bir deneysel çalışma yapılmıştır. Deneysel çalışmada üretim ve çekme kanban sayıları eşit alınmış ve kanban sayısının 1 ile 30 arasındaki değerlerinin her birisi için benzetim modeli çalıştırılarak toplam maliyet değeri tahmin edilmiştir. Toplam maliyet değerinin tahmininin önce azaldığı sonra sürekli arttığı görülmüştür. Dolayısıyla, maliyet değerlerinin birbirine yakın olduğu 1-6 aralığı tüm kanbanlar için alt ve üst düzey olarak seçilmiştir.

**Tavlama Benzetimi:** Bu çalışmada benzetim eniyilemesi amacıyla hazırlanan TB algoritması Şekil 5'de verilmektedir. TB algoritmasında başlangıç sıcaklığının, sıcaklık azaltma oranının, her sıcaklıktaki tekrar sayısının ve durdurma koşulunun belirlenmesi "tavlama" veya "soğutma planı" olarak tanımlanmaktadır. Bu planın seçimi algoritmanın

İstasyonlar	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Üretim Kanbanı	2	3	1	3	5	6	3	2	3	4	5	2	1	6
İstasyonlar arası	1-5	2-6	3-7	4-8	5-9	6-10	7-12	8-12	9-11	10-11	11-12	12-13	13-14	
Çekme Kanbanı	2	6	1	2	2	2	3	4	6	2	6	3	4	-

Şekil 4. Dizi Kodlaması

etkinliği açısından önemlidir. Bu çalışmada yapılan ön denemeler sonucunda, başlangıç sıcaklığı için 1.000.000 ve 4.000.000, azaltma oranı için 0.10, 0.05 ve 0.03, aynı sıcaklık düzeyinde aranan nokta sayısı için 10 ve 50, aday nokta sayısı için 1 ve 5 değerleri olmak üzere 24 parametre kombinasyonu ve iki farklı hareket mekanizması dikkate alınmıştır. Birinci hareket mekanizmasında rassal seçilen bir üretim ve bir çekme kanbanına rassal birer değer verilmiştir. Diğer hareket mekanizmasında ise rassal olarak seçilen bir üretim ve bir çekme kanbanının değerleri rassal olarak 1 artırılmış veya 1 azaltılmıştır. Toplam 48 farklı kombinasyonla yapılan denemelere göre eniyi yakınsamayı veren başlangıç sıcaklığı 4.000.000, aday nokta sayısı 1, her sıcaklıkta aranan nokta sayısı 50, sıcaklık azaltma oranı 0.03 ve hareket mekanizması için rassal seçilen bir üretim ve bir çekme kanbanına rassal birer değer verilmesi kombinasyonu seçilmiştir. Son sıcaklık değeri ise yapılan ön çalışma sonucunda 165000 olarak belirlenmiştir.

**Genetik Algoritmalar:** TZÜ sisteminde kanban sayısının eniyilenmesi için kullanılan GA, Şekil 6'da verilmektedir. Bir GA'nın performansında yığın genişliği ile çaprazlama ve mutasyon operatörlerinin kullanım olasılığı etkili olmaktadır. Bu nedenle, çalışmada öncelikle bu parametreler için en uygun kombinasyon belirlenmeye çalışılmıştır. Bu amaçla, yığın genişliği için 20 ve 50, çaprazlama olasılığı için 0.60, 0.75 ve 0.90, mutasyon olasılığı için 0.10, 0.15 ve 0.20 olasılıkları yapılan ön denemelere dayalı olarak dikkate alınmıştır. 18 farklı parametre kombinasyonu, iki farklı mutasyon operatörü için denemeler yapılmıştır. Birinci mutasyon operatöründe, rassal olarak belirlenen bir genin değeri 1-6 arasında rassal olarak değiştirilmiştir. İkinci mutasyon operatöründe ise rassal olarak belirlenen bir genin değeri, 0.50 olasılıkla bir artırılmış veya bir azaltılmıştır. Toplam 36 farklı kombinasyonla yapılan denemelere göre, yığın genişliği 50, çaprazlama oranı 0.90, mutasyon oranı 0.20 ve 1-6 arasında rassal olarak çalışan mutasyon operatörü seçilmiştir. GA'da, tek

---

**Adım1:** Başlangıç çözümünü ( $x_0$ ) rassal olarak üret.

Yeni çözüme ( $x_y$ ) başlangıç çözümünü ata:  $x_y = x_0$

$x_y$  için amaç fonksiyonu ( $f(x_y)$ ) değerini benzetim modeli ile tahmin et.

$x^* = x_y$  ve  $f(x^*) = f(x_y)$

$T = T_0$ ,  $r = 0.97$

**Adım2:**  $k = 0$

**Adım3:** Yeni çözümden rassal olarak bir üretim ve bir çekme kanbanı seç, bunlara yeni rassal değerler vererek aday çözümünü ( $x_a$ ) üret ve ( $f(x_a)$ ) değerini benzetim modeli ile tahmin et.

$I = f(x_a) - f(x_y)$

$k = k + 1$

Eğer  $I < 0$  ise  $x_y = x_a$  değilse  $u \sim U(0,1)$  üret.

Eğer  $u < \exp\left[-\frac{I}{T}\right]$  ise  $x_y = x_a$ , değilse adım 5'e git.

**Adım4:** Eğer  $f(x_y) < f(x^*)$  ise  $x^* = x_y$  ve  $f(x^*) = f(x_y)$

**Adım5:**  $k = k_s$  ise Adım6'ya git, değilse Adım3'e git.

**Adım6:**  $T = T * r$

**Adım 7:**  $T < T_s$  ise dur, değilse Adım2'ye git.

---

**Şekil 5.** Kanban Eniyilemesi için TB Algoritması

---

**Adım1:**  $m=0$ ,

$n$  diziden oluşan başlangıç yığınınını rassal olarak oluştur.

Her dizi için amaç fonksiyonu değerini ( $f(x)$ ) benzetim modelini kullanarak tahmin et.

Maliyet değerlerini kullanarak her dizinin uygunluk değerini hesapla.

**Adım2:** Başlangıç yığınındaki amaç fonksiyonu değeri enküçük olan diziyi ( $x_{meç}$ ) eniyi çözüm ( $x^*$ ) olarak seç.  $x^* = x_{meç}$  ve  $f(x^*) = f(x_{meç})$

**Adım3:** Rulet çemberi yöntemini kullanarak geçici yığını oluştur.

**Adım4:** Geçici yığına çaprazlama ve mutasyon operatörlerini

( $p_{cap}$  ve  $p_{mut}$  oranları ile) uygulayarak yeni yığını oluştur.

**Adım5:** Yeni yığındaki her dizi için amaç fonksiyonu değerini ( $f(x)$ ) benzetim modeli ile tahmin et ve  $x_{meç}$ 'i belirle.

**Adım6:**  $x_{meç} < x^*$  ise  $x^* = x_{meç}$  ve  $f(x^*) = f(x_{meç})$

**Adım7:**  $m = m + 1$ ,  $m > 100$  ise dur değilse adım 3'e git

---

**Şekil 6.** Kanban Eniyilemesi için GA Algoritması

noktalı çaprazlama operatörü ve rulet çemberi seçim mekanizması kullanılmıştır. Seçim mekanizması olarak rulet çemberi kullanıldığından dolayı, problem enbüyüklemeye problemine dönüştürülerek probleme çözüm aranmıştır. Mevcut yığındaki enbüyük maliyetten diğer diziler maliyetleri çıkarılarak problem enbüyüklemeye

problemine dönüştürülmüştür. Algoritma, istenilen iterasyon sayısına ulaştığında durmaktadır.

**Tabu Arama:** Bu çalışmada benzetim eniyilemesi amacıyla hazırlanan TA algoritması Şekil 7'de verilmektedir. Geliştirilen TA algoritmasında her iterasyonda artırma/azaltma komşuluk yapısı

---

**Adım1:**  $m = 0$

Başlangıç çözümünü ( $x_0$ ) rassal olarak üret ve amaç fonksiyonu ( $f(x_0)$ )değerini benzetim ile tahmin et. Başlangıç çözümünü, mevcut çözüm ve eniyi çözüm olarak al.

$x_{mev} = x_0, x^* = x_0$  ve  $f(x^*) = f(x_0)$

Tabu hareketleri listesini sıfırla.

**Adım2:**  $k = 1, 2, \dots, 54$  için  $x_{mev}$ 'in tüm komşularını ( $x_k$ ) komşuluk yapısını kullanarak üret.

$k = 1, 2, \dots, 54$  için  $f(x_k)$  değerini benzetim ile tahmin et.

Tabu olmayan bir hareketle elde edilmiş ya da tabu yıkma kriterini sağlamış olan amaç fonksiyonuna göre eniyi komşuyu bul ( $x_{ek}$ ) ve mevcut çözüm olarak kabul et ( $x_{mev} = x_{ek}$ ).

**Adım3:**  $m = m + 1$

$f(x_{ek}) < f(x^*)$  ise  $x^* = x_{ek}$  ve  $f(x^*) = f(x_{ek})$

$m > 100$  ise dur. Değilse adım 4'e git

**Adım4:** Tabu hareketleri listesini güncelle ve Adım2'ye git.

---

**Şekil 7.** Kanban Eniyilemesi için TA Algoritması

kullanılarak 27 kanbandan oluşan bir çözümden 54 tane komşu çözüm üretilmektedir. Bu komşuluk yapısında bir kanban tipinin aldığı değer bir birim artırılırken/azaltılırken diğer kanban tiplerinin sayıları sabit tutulmaktadır. Böylece algoritma kanban tipi sayısının 2 katı kadar komşu çözüm ile çalışmaktadır. Bu komşular arasından hesaplanan eniyi amaç fonksiyonu değerine sahip olan hareket tabu değilse veya tabu yıkma kriterini sağlıyorsa yeni mevcut çözüm olarak seçilmektedir. Bir hareketten sonra, değeri değişen kanban ve bu kanbanın eski değeri tabu listesine eklenmektedir. Algoritmanın hangi tabu süresinde daha iyi performans gösterdiğini bulmak için literatürde önerilen 7-12 arasındaki değerler dikkate alınarak denemeler yapılmış ve eniyi yakınsamayı gösteren tabu süresinin 10 olduğu görülmüştür. İstenilen en büyük iterasyon sayısına ulaşıldığında ise algoritma durdurulmaktadır.

### ALGORİTMALARIN KARŞILAŞTIRILMASI

Bu çalışmada, düşünsel olarak gerçeğe uygun şekilde tanımlanan büyük boyutlu, karmaşık bir TZÜ sisteminde eniyi kanban kombinasyonunun belirlenmesi amacıyla kullanılan arama algoritmaları TB, TA ve GA çözüm kalitesi ve iyi çözüme yakınsama hızı açısından karşılaştırılmıştır. Ayrıca, bu algoritmaların performansı rassal arama (RA) algoritmasına göre de değerlendirilmiştir. RA algoritması, çözüm uzayında sadece rassal olarak arama yaparak amaç fonksiyonunu eniyilemeye çalışan bir sezgisel arama algoritmasıdır. Bu çalışmada kullanılan RA algoritmasında, belirlenen durdurma koşulu sağlanıncaya kadar rassal olarak seçilen kanban kombinasyonları TZÜ sisteminin benzetim modeline amaç fonksiyonu değerini hesaplayabilmek için girdi olarak verilmiştir. Amaç fonksiyonunu eniyileyen yani toplam maliyeti en küçük olan kanban kombinasyonu problemin çözümü olarak seçilmiştir. Her algoritma bir denemede 5000 nokta incelemek üzere 3 kez çalıştırılmıştır. Her tekrarda toplam 8000 sipariş karşılanıncaya kadar benzetim modeli çalıştırılmak üzere her noktada 5 tekrar yapılmıştır.

Tablo 2'de bu çalışmada kullanılan algoritmaların, enküçük maliyet, ortalama maliyet, değişim katsayısı, eniyi çözüme ulaşmak için aradıkları çözüm sayısı açısından karşılaştırılmaları verilmektedir. Tablo 2 incelendiğinde TB algoritması 33436156 pb ile enküçük maliyeti bulurken GA da 35481876 pb ile ikinci enküçük maliyeti bulabilmektedir. Ayrıca, GA ve TB' nin 3 deneme için değişim katsayılarının sırası ile 0.030 ve 0.042 değerleri ile diğer iki arama algoritmasına göre daha düşük olduğu görülmektedir. Bu durum, GA ve TB'nin iyi çözümlere ulaşmada başlangıç çözümlerine karşı duyarsız (robust) olduğunu göstermektedir. RA ise beklenen şekilde diğer 3 algoritmaya göre enbüyük maliyeti bulmuştur. Bu çalışmada dikkate alınan problemin çözüm uzayı 6 adet noktadan oluşmaktadır. Algoritmaları iyi çözüme yakınsamaları açısından karşılaştırıldıklarında TA, TB, GA ve RA sırası ile ortalama 150, 3250, 4200 ve 4800 noktayı inceledikleri görülmektedir. Algoritmaların 3 deneme sonucuna göre ortalama yakınsama hızları Şekil 8'de verilmektedir. Şekil 8 incelendiğinde, TA en hızlı yakınsayan algoritma iken ortalama eniyi çözümü veren algoritmaların ise TB ve GA olduğu görülmektedir. Bilindiği gibi her algoritmada bir kanban kombinasyonu için maliyet değerinin hesaplanabilmesi için öncelikle benzetim modeli kullanılması gerekmektedir. Benzetim modeli ile elde edilen değerler tahmin değerleridir. Bu nedenle, TB ve GA ile bulunan ve enküçük maliyeti veren kanban kombinasyonları ile çalışan iki sistem arasında anlamlı bir farklılık olup olmadığı istatistiksel olarak incelenmiştir. %95 güvenlik düzeyinde güven aralığı [-24.382.787 , 9.290.674] olarak bulunmuştur. Bu durumda, GA ve TB algoritmaları ile bulunan kanban kombinasyonları ile çalışan iki sistem arasında anlamlı bir farklılık yoktur. Ancak, aranan çözüm sayısı kadar benzetim modelinin çalıştırılması gereği dikkate alındığında TB algoritmasının sağlayacağı hesaplama zamanı avantajı benzetim eniyilemesinin yapıldığı bu çalışmada önemli kazanç sağlamıştır. TZÜ için TB algoritması ile elde edilen eniyi kanban kombinasyonu Şekil 9'da verilmektedir.

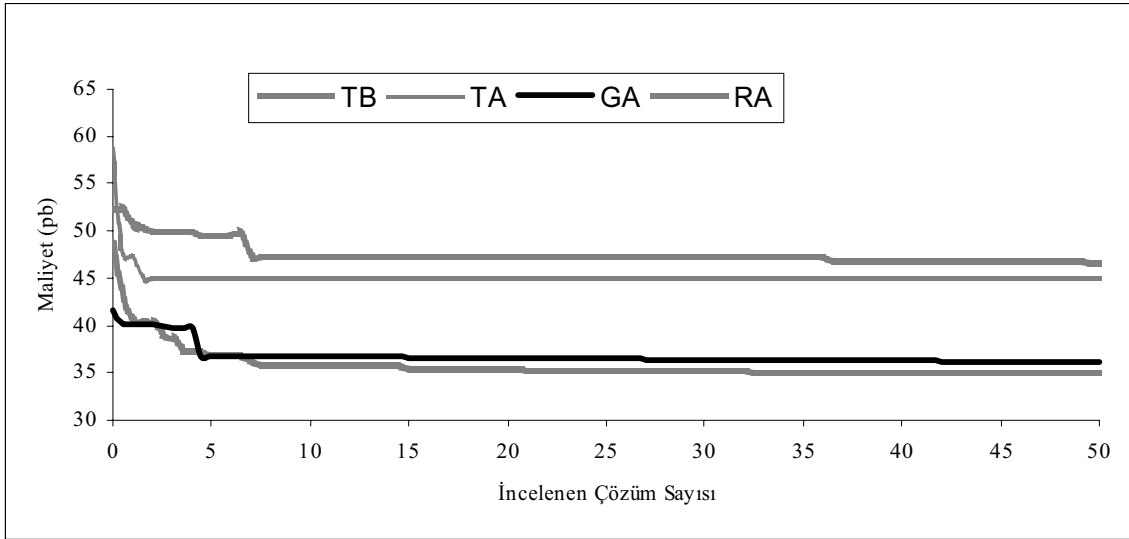
## SONUÇ

Bu çalışmada tam zamanında üretim sistemlerinde eniyi kanban kombinasyonunun bulunması amacıyla genel amaçlı arama algoritmaları kullanılmıştır. Sistemin benzetim modeli SIMAN benzetim dili ile modellenmiş ve C programlama dili ile kodlanan genel amaçlı arama algoritmaları ile bütünleşik kullanılmıştır. Benzetim eniyilemesi için Rassal Arama (RA), Tabu Arama (TA), Genetik Algoritma (GA) ve Tavlama

Benzetimi (TB) algoritmaları kullanılmıştır. Çözüm kalitesi açısından algoritmalar karşılaştırıldığında TB'nin eniyi çözümü verdiği görülmüştür. Ayrıca ilk iki eniyi çözümü veren TB ve GA arasında yapılan istatistiksel karşılaştırma sonucunda her iki çözüm arasında 0.05 anlamlılık düzeyinde anlamlı bir farklılık olmadığı yani her iki algoritmanın çözüm kalitesinin eşdeğer olduğu gösterilmiştir. TB ve GA iyi çözümlere sırası ile ortalama 3250 ve 4200 nokta inceleyerek ulaşmaktadır. Bu iki algoritma arasında çözüm kalitesi açısından anlamlı

**Tablo 2.** Algoritmaların Karşılaştırılması

	TB	TA	GA	RA
<b>Enküçük Maliyet (pb)</b>	33436156	42035944	35481876	<b>43030050</b>
<b>Ortalama Maliyet (pb)</b>	35146133	45021110	36132709	46592672
<b>Değişim Katsayısı</b>	0.042	0.100	0.030	<b>0.126</b>
<b>Aranan Çözüm Sayısı</b>	3250	150	4200	<b>4800</b>
<b>Arama Yüzdesi</b>	$3.175 \times 10^{-16}$	$1.466 \times 10^{-17}$	$4.104 \times 10^{-16}$	



**Şekil 8.** Algoritmaların Yakınsama Grafiği

İstasyonlar	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Üretim Kanbanı	5	1	2	1	3	1	4	6	1	3	1	5	2	6
İstasyonlar arası	1-5	2-6	3-7	4-8	5-9	6-10	7-12	8-12	9-11	10-11	11-12	12-13	13-14	
Çekme Kanbanı	1	1	2	3	3	3	5	2	6	6	2	3	3	-

**Şekil 9.** TZÜ için TB ile Bulunan Eniyi Kanban Kombinasyonu

bir farklılık olmamasına rağmen her bir çözüm noktası için amaç fonksiyonunun değeri hesaplanmasında benzetim modelinin kullanılması gereği çözüm zamanı açısından büyük bir önem taşımaktadır. Dolayısıyla bu çalışmada benzetim eniyilemesi amacı ile kullanılan dört algoritma arasından eniyi performansı TB algoritması göstermiştir.

### KAYNAKÇA

1. Ahmed, M. A. And Alkhamis, T. M., 1998, "Simulation Based Optimization Using Simulated Annealing with Ranking and Selection", *Computers and Operation Research*, 387-402
4. Ahmed, M. A. And Alkhamis, T. M. and Hasan M., 1997, "Optimizing Discrete Stochastic Systems Using Simulated Annealing and Simulation", *Computers and Operation Research*, 823-836
5. Azadivar, F., Tompkins, G., 1999, "Simulation Optimization with Qualitative Variables and Structural Model Changes: A Genetic Algorithm Approach", *European Journal of Operational Research*, 113(1), 169-182
6. Altıparmak, F., 1996, "Genetik Algoritma ile Haberleşme Şebekelerinin Topolojik Optimizasyonu", Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı Doktora Tezi, 36-63
7. Alabaş, Ç., 1999, "Tabu Arama ve Tavlama Benzetimi Algoritmalarıyla Bilgisayar Şebekelerinin Topolojik Optimizasyonu", Gazi Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 29-56
8. Alabaş, Ç., Altıparmak, F., Dengiz, B., 2002, "A Comparison of the Performance of Artificial Intelligence Techniques for Optimizing the Number of Kanbans", *Journal of the Operational Research Society*, 53, 907-914.
9. Banks, J., Carson, J.S, Nelson, B.L, Nicol, D.M, 2001, *Discrete-Event System Simulation*, Prentice-Hall, 485-492
8. Bulgak, A.A., and Sanders, J.L., 1988, Integrating a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems, *Proceedings of the 1988 Winter Simulation Conference*, 684-690.
9. Dengiz, B., Altıparmak, F., 1998, "Genetik Algoritmalarla Genel Bir Giriş", *Endüstri Mühendisliği Dergisi*, 9(3), 3-10
10. Dengiz, B., Altıparmak, F., Bulgak, A., 1998, "Optimization of Stochastic Systems Using Genetic Algorithms", *Yöneylem Araştırması Dergisi/Transactions on Operational Research*, 9(10), 43-62
11. Fukukawa, T., Hong, S.C., 1991, "The determination of the optimal number of kanbans in a JIT production system", *Computers and Industrial Engineering*, 551-559.
12. Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1-88.
13. Gupta, Y., Gupta, M., 1989, "A system dynamics model for a multi-stage multi-line dual-card JIT-kanban system", *International Journal of Production Research*, 27(2), 309-352.
14. Haddock, J., Mittenthal, J., 1992, "Simulation Optimization Using Simulated Annealing", *Computers and Industrial Engineering*, 387-395
15. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., 1953, "Equation of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, 21, 1087-1092.
16. Michalewicz, Z., 1992, "Genetic Algorithms+Data Structures=Evolution Programs", Springer Verlag Berlin Heidelberg, 1-53
17. Nahmias, S., 1993, *Production and Operations Analysis*, 2nd Edition, Irwin, USA.
18. Pierreval, H., Paris, J.L., 2000, "Distributed Evolutionary Algorithms for Simulation Optimization", *IEEE Transactions on Systems, Man and Cybernetics*, 30(1), 15-24
19. Pierreval, H., Paris, J.L., 2001, "A Distributed Evolutionary Simulation Optimization Approach for the Configuration of Multiproduct Kanban Systems", *International Journal of Computer Integrated Manufacturing*, 14(5), 421-430
20. Pierreval, H., Tautou, L., 1997, "Using Evolutionary Algorithms and Simulation for the Optimization of Manufacturing Systems", *IIE Transactions*, 29, 181-189
21. Paul, R.J., Chaney, T.S., 1998, "Simulation Optimization Using a Genetic Algorithm", *Simulation Practice and Theory*, 6, 601-611
22. Rees, L.P., Philipoom, P.R., Taylor, B.W., Kuang, P.Y., 1987, "Dynamically adjusting the number of kanbans in JIT production system using estimated values of leadtime", *IIE Transactions*, 19(2), 199-207.
23. Wang, H., Wang, H., 1991, "Optimum number of kanbans between two adjacent workstations in a JIT system", *International Journal of Production Economics*, 22, 179-188.
24. Yunker, J.M., Tew, J.D., 1994, "Simulation Optimization by Genetic Search", *Mathematics and Computers in Simulation*, 37, 17-28