

YAPIM PROJELERİNDE GENETİK ALGORİTMA KULLANARAK KAYNAK SEVİYELEME

Mustafa ORAL

M.K.Ü., Elektrik-Elektronik Mühendisliği Bölümü, Hatay/Türkiye

Emel LAPTALI ORAL

M.K.Ü., İnşaat Mühendisliği Bölümü, Hatay/Türkiye

Semra BOZKURT

Eskici Gayrimenkul, Adana/Türkiye

Ercan ERDİŞ

Ç.Ü., İnşaat Mühendisliği Bölümü, Adana/Türkiye

ÖZET : *Bu çalışma kapsamında, yapım projelerinde kaynak seviyeleme problemlerinin çözümü için genetik algoritma mantığına dayanarak çözüm geliştiren bir bilgisayar yazılımı geliştirilmiş ve validasyonu yapılmıştır. Bu yazılımda, literatürde tartışılan programlara ek olarak üç tip çeşitlendirme yöntemi denenmiş, normalleştirmenin en hızlı sonuçları verdiği gözlenmiştir.*

RESOURCE LEVELING IN CONSTRUCTION PROJECTS BY USING GENETIC ALGORITHMS

ABSTRACT : *Within the context of this study, a computer software that is based on genetic algorithm logic has been developed and validated for solving resource leveling problems in construction projects. In addition to the literature findings, three different types of scaling has been included within the software. Normalizing Scaling has been determined to be providing the fastest results.*

1. GİRİŞ

Yapım projelerinin hedeflenen süre, maliyet ve kalitede tamamlanması, projeyi yürütmek için gereken kaynakların yapım süreci öncesi gerçekçi analiz ve planlaması ile mümkündür. Kaynak analizi/planlaması, eldeki kaynakların projenin ilgili faaliyetlerine projenin süre/maliyet kısıtları göz önünde tutularak tahsis edilmesini içerir. Bu süreçte dikkat edilmesi gereken diğer önemli unsur ise kaynak seviyelerinin proje süresince tüm birim zamanlarda aynı seviyede tutulmasıdır.

Geleneksel analitik ve heuristik yaklaşımlar yapım projeleri için kaynak seviyeleme problemlerinin çözümünde yetersiz ve esneklikten uzak kalmaktadırlar (1). Genetik Algoritmalar (GA'lar) kaynak seviyeleme problemini geleneksel yöntemlerden çok daha hızlı ve optimuma yakın olarak çözmektedirler (2). Bunun en önemli nedeni problemin yapısının GA'ya uygun olmasıdır; ulaşılmak istenen kaynak seviyesi bilinmekte, bu çözüme en yakın değerleri verecek çözüm seti aranmaktadır.

Bu çalışma kapsamında kaynak seviyeleme problemlerinde GA kullanarak optimizasyon yapan bir yazılım geliştirilmiştir. Elde edilen sonuçlar literatür bulguları ile beraber tartışılmaktadır.

2. KAYNAK SEVİYELEME

Kaynak seviyelemede hedef, projenin süre ve maliyet kısıtları göz önünde tutularak, birim zamanda kullanılacak kaynak miktarının proje süresi boyunca Ortalama Kaynak Düzeyi (OKD)'nde seyretmesinin sağlanmasıdır. Bu da OKD'den sapmaların minimize edilmesi ile mümkündür. Kaynak seviyeleme şu aşamaları içerir:

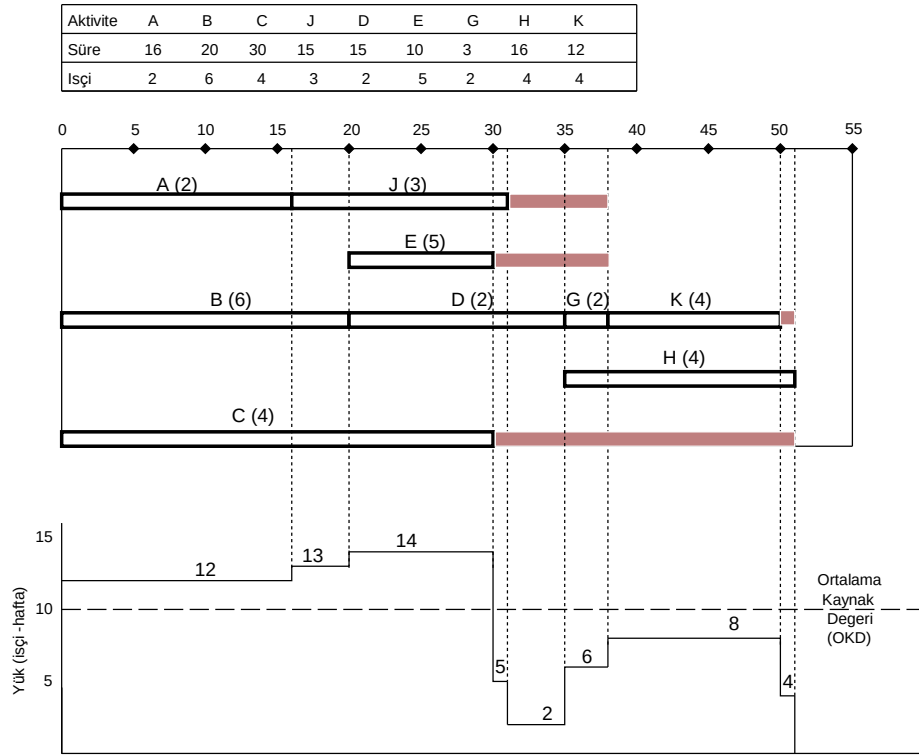
- (1) Kaynak dağıtımının yapılması: Projenin planlanan sürede bitirilebilmesi için her aktiviteye gerekli olan kaynak miktarlarının belirlenmesi aşamasıdır. Belirlenen kaynak miktarlarına ve aktivite sürelerine göre projenin iş programı ve kaynak histogramı çıkarılır (Şekil 1a).
- (2) OKD değerinin belirlenmesi: OKD değeri eşitlik (1)'den elde edilebileceği gibi, eldeki kaynakların kısıtlı bir miktarda olması durumunda bu miktara eşit olarak da alınabilir.

$$OKD = \frac{\sum_{i=1}^n x_i k_i}{T} \quad (1)$$

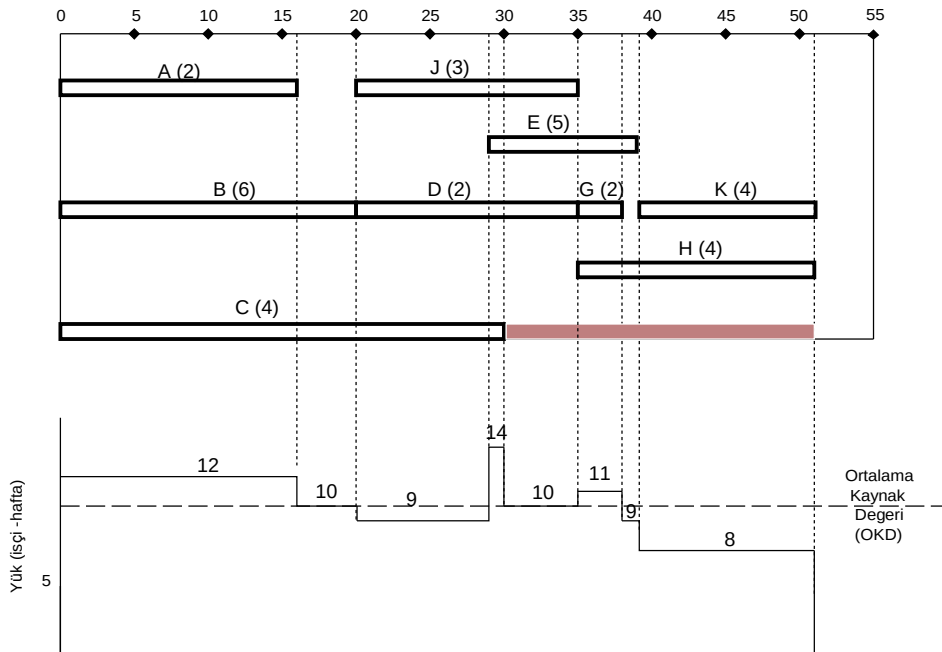
X_i i. aktivite süresi, k_i i. aktivitesi için birim zamanda gereken kaynak miktarı, n projedeki aktivitelerin sayısı, ve T toplam proje süresini belirtmektedir.

- (3) OKD değerine göre kaynak seviyelemenin yapılması: Bu aşamada kaynak dağıtımı OKD değerine göre yapıldığında proje süresinin uzayıp uzamayacağı analiz edilir. Proje süresinin uzaması istenmiyorsa bazı zaman dilimlerinde kullanılan kaynak miktarı OKD'den sapabilecektir (Şekil 1b).

YAPIM PROJELERİNDE GENETİK ALGORİTMA KULLANARAK KAYNAK SEVİYELEME



Şekil 1a. Örnek proje kaynak seviyeleme öncesi iş programı ve kaynak histogramı



Şekil 1b. Örnek proje kaynak seviyeleme sonrası iş programı ve kaynak histogramı

3. GENETİK ALGORİTMA

Genetik Algoritma (GA), bir veri grubundan özel bir veriyi bulmak için kullanılan bir arama metodudur. Genetik algoritmalar doğada geçerli olan en ‘iyi’lerin hayatta kalması kuralına dayanarak bir çözüm grubu içerisinde sürekli iyileşen çözümleri arar. Burada ‘iyi’, problemin özelinde belirlenmiş kriterlere uymak anlamındadır. ‘İyi’ lerin seçimi için bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme gibi operatörler kullanır.

Belirli bir problemde GA kullanarak en iyi çözümü aramak için öncelikli olarak problemin parametrelerinin (değişkenler ve kısıtlar) belirlenmesi gerekir. Değişkenlerin her birisi, ‘bireyi’ oluşturan kromozom üzerindeki bir gene tekabül etmektedir. Gen sayısının dikkatsiz seçimi çözüm uzayının boyut sayısını arttıracaktır. Ayrıca her bir gen, kısıtlara bağlı olarak belirli bir değer aralığına sahiptir. Değer aralığının büyüklüğü çözüm uzayının genişlemesine ve dolayısı ile aranan çözüme erişim zamanının artmasına neden olacaktır. Bu yüzden değer aralığı minimum seviyede tutulmalıdır. Uygun sayıda ve uygun aralıkta tutulmuş genlerin bir araya gelmesi ile oluşan her bir kromozom, çözüm uzayında bir noktadır. Optimum çözüm arayışına, bu uzaydan rasgele seçilmiş bireylerden oluşan bir başlangıç topluluğu ile başlanır.

Topluluk içerisindeki bireyler, genlerini bir sonraki kuşağa aktarmak için, öncelikle uygunluk değerlerine göre seçilirler ve ikiye bölünerek ürerler. Seçilme aşamasında topluluktaki bireyler birbirleri ile yarışır. Topluluktaki çoğu bireyin uygunluklarının aynı olması durumunda topluluğun içerisinde bireylerin birbiriyle yarış güçleşmekte ve kuşakların daha iyiye doğru gidişi zorlaşmaktadır. Bu aşamada çeşitlendirme, seçimden önce devreye girer. Çeşitlendirme yapılarak kromozomların bağlı uygunluk değerleri arasında kabul edilebilir bir fark oluşturulur (3). Genelde üç tip çeşitlendirme vardır: Doğrusal, Normalleştirme ve Dinamik Doğrusal. Çeşitlendirme sonrasında uygun kromozomların seçimi Rulet Tekerleği veya Turnuva Seçimi gibi olasılık hesaplarını temel alan yöntemler ile yapılır. Seçim de başlangıç topluluğunun oluşturulması gibi rasgeledir. Ancak bu rasgele seçimde seçilme olasılıklarını çözümlerin uygunluğu belirler.

Eşleşme hakkını kazanmış bireyler kromozomlarının rasgele bir noktadan kesilerek çaprazlanması ile geçici topluluğun yeni bireylerini oluştururlar. Çaprazlama işlemi tek noktadan olduğu gibi birden fazla noktadan da yapılabilir. Bazı durumlarda iki ebeveyn kromozom da çaprazlama olmadan geçici topluluğa girebilir. Geçici topluluğun önceki topluluğun aynısı olma olasılığını ortadan kaldırmak için kromozomlardaki bazı gen değerleri belli bir mutasyon olasılığına göre değiştirilir ve yeni topluluk oluşturulur.

Üreme, çaprazlama ve mutasyon işlemleri sonrasında bir önceki toplulukta bulunan en uygun birey yeni topluluğa aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra oluşan yeni topluluğa bir önceki topluluğun en uygun (elit) bireyi herhangi bir birey ile değiştirilerek eklenir (4).

Yukarıda belirtilen işlemler en son elde edilen topluluğun başlangıç topluluğu olarak alınması ile tekrarlanır. Her bir döngü bir jenerasyona denk düşer. Algoritmanın ne zaman sonlanacağına kullanıcı karar verir. GA’nın belli bir sonlanma kriteri yoktur. Yeni kuşaktaki en uygun bireyin uygunluk fonksiyonu değerinin belli bir değere yakınsaması veya algoritmanın belli sayıda döndürülmesi, algoritmanın durması için kriter olarak kullanılabilir.

4. ÖNCEKİ ÇALIŞMALAR

Literatürde, kaynak seviyeleme problemlerinin çözümünde Tamsayılı Programlama (5), Dışavurumlu Sayımlama (6), Minimum Moment Algoritması (7) ve heuristic yöntemlere dayanan optimizasyon metotları (8,9) kullanılmıştır. Genetik algoritma kullanılarak çözümlenmesi yönünde ise 4 önemli çalışma bulunmaktadır.

AL-TABTABAI ve ALEX (10) genel olarak GA'nın yapım işlerindeki optimizasyon problemlerinde nasıl faydalı olacağını tartışmaktadırlar. GA'nın fayda sağlayacağı bir alan da kaynak seviyeleme olarak belirtilmektedir. Tamsayılı programlama ile çözülmüş bir kaynak seviyeleme problemi GA ile de çözülmüş ve karşılaştırma yapılmıştır.

HEGAZY (9) ise kaynak yerleştirme ve seviyeleme optimizasyonunda GA'yı kullanmıştır. Yazarın yaklaşımı ile ilk defa kaynak yerleştirme ve seviyeleme için aynı anda optimum aranmaktadır.

OGWU ve TAH (2) ise bir hibrit genetik algoritma yaklaşımında bulunmaktadır. Çalışmalarında GA ile organizasyon veri tabanlarını birleştirmişlerdir.

LEU ve ark. (1) ise GA tabanlı bir karar destek sistemi geliştirmişlerdir. Problem çözümünde kurdıkları amaç fonksiyon bu makalenin konusunu oluşturan çalışmada temel alınmıştır.

5. MATERYAL METOD

GA ile kaynak seviyelemeye amaç proje süresince birim zamanlarda gereken kaynak miktarı seviyelerindeki dengesizliği minimuma indirmektedir. Şekil 1'de de gösterildiği gibi aktiviteler toplam bollukları da göz önünde tutularak sürüklenirildiklerinde proje birim zamanında gerekli kaynak miktarları da değişmektedir.

Bu çalışmada GA kullanarak, aktiviteleri bölünmeyen, aktivite süreleri, ilişkileri ve aktivitelerinin kaynak talebi sabit olan projelerde tek tip kaynak seviyelemeyi hedefleyen bir yazılım geliştirilmiştir. Bu amaçla aşağıda sıralanan parametreler kullanıcı tarafından programa girilir:

- | | |
|----------------------------------|---------------------------|
| (a) projenin aktivite sayısı | (b) proje süresi |
| (c) topluluk büyüklüğü | (d) maksimum kuşak sayısı |
| (e) çaprazlama olasılığı | (f) mutasyon olasılığı |
| (g) çeşitlendirme tipi | (h) çeşitlendirme sabiti |
| (i) elitizm istenip istenmediği. | |

Problemin çözümünde kullanılan GA için kromozomlar oluşturulurken her bir genin değeri (2) no'lu eşitsizlik kısıtlaması çerçevesinde aktivitenin başlama zamanı olarak atanmıştır (Şekil 2).

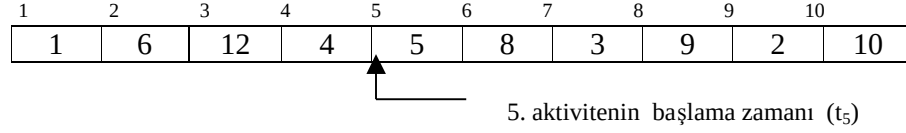
$$t_i - EB_i \leq TB_i, \quad t_i \geq 0, \quad i = 1, 2, 3, \dots, g \quad (2)$$

t_i = her i aktivitesinin başlama zamanı

EB_i = her i aktivitesinin en erken başlama zamanı

g = aktivite (gen) sayısı

TB_i = her i aktivitesinin toplam bolluk süresi



Şekil 2. Kromozom tanımlaması

Çaprazlama için tek nokta seçilmiş ve genler düzenli olarak mutasyona uğratılmıştır. Seçim yöntemi olarak Rulet Tekerleği metodu kullanılmıştır. Seçim için gerekli olan uygunluk fonksiyonu proje süresince kaynak seviyelerinin OKD'den sapmalarının minimum olmasını gerektirir. Böylelikle problemin uygunluk fonksiyonu 'z' için eşitlik (3)'deki gibi olur.

$$\text{Min } z = \sum_{q=1}^T \left(\sum_{i=1}^g |r_{iq} - \text{OKD}| \right) \quad (3)$$

i = aktivite
T = proje süresi
r_{iq} = i aktivitesinin q. birim zaman diliminde gerektirdiği kaynak miktarı

q = birim zaman dilimi
g = aktivite sayısı

Bunlara ek olarak, çalışmada üç ayrı tip çeşitlendirme yönteminin (Tablo 1) kullanımına imkan sağlanmıştır.

Tablo 1. Çalışmada kullanılan çeşitlendirme yöntemleri

Doğrusal	Dinamik Doğrusal	Normalleştirme
$f^1_k = a \cdot f_k + b$	$f^1_k = a \cdot f_k + b_t$	$f^1_k = \frac{f_k - f_{\min} + \gamma}{f_{\max} - f_{\min} + \gamma}$
$a = \gamma \cdot [MF / [MF - f_{\max}]]$	$a = \gamma \cdot [MF / [MF - f_{\max}]]$	
$b = (1 - a) \cdot MF$	$b_t = -f_{\min}$	
f^1_k : k.ncı kromozomun yeni uygunluk değeri f_k : k.ncı kromozomun ilk uygunluk değeri MF : Topluluktaki bütün bireylerin ortalama uygunluğu $f_{\max}$: Topluluktaki en büyük uygunluk γ : Çeşitlendirme Sabiti		

Programın validasyonu için uygulanan örnek problemlerden birisi ve kullanılan parametre değerleri Tablo 2 ve 3'de verilmektedir. Tablo 2'de verilen aktivite ilişkileri ve süreleri programa girilmiş, program öncelikle kaynak seviyelemesini yaptıktan sonra, üç tip çeşitlendirmeden hangisinin daha hızlı sonuç verdiği karşılaştırılmıştır.

Tablo 2. Örnek problemi oluşturan aktivitelerin başlama ve bitiş tarihleri ve toplam bollukları

Aktivite	Süre	Erken baş.	Erken bit.	Geç baş.	Geç bit.	Toplam bolluk
A	2	0	2	6	8	6
B	4	0	4	0	4	0
C	1	0	1	3	4	3
D	4	2	6	10	14	8
E	3	4	7	8	11	4
F	6	4	10	4	10	0
G	6	1	7	4	10	3
H	1	7	8	14	15	7
J	4	7	11	11	15	4
K	5	10	15	10	15	0
L	1	15	16	15	16	0

Tablo 3. Örnek problemde kullanılan parametreler

Parametre	Değer
Aktivite sayısı	11
Proje süresi	15
Populasyon büyüklüğü	100
Her genin bit uzunluğu	12
Maks. jenerasyon sayısı	300
Çaprazlama oranı	0.6
Mutasyon oranı	0.4
Çeşitlendirme katsayısı	0.5
Elitizm	Açık

Program ile seviyeleme yapılmadan önceki kaynak kullanımı ve çubuk diyagramı Şekil 3’de, seviyeleme sonrası kaynak kullanımı ve çubuk diyagramı ise Şekil 4’de verilmektedir.

Program					Zaman															
Akt	Kay	Baş	Bitiş	Süre	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	2	0	2	2	■	■														
B	1	0	4	4	■	■	■	■												
C	4	0	1	1	■															
D	4	2	6	4			■	■	■	■										
E	2	4	7	3					■	■	■									
F	4	4	10	6					■	■	■	■	■	■						
G	6	1	7	6		■	■	■	■	■	■									
H	0	7	8	1								■								
J	2	7	11	4								■	■	■	■					
K	1	10	15	5											■	■	■	■	■	■
L	2	15	16	1												■	■	■	■	■
Günlük kaynak kullanımı					7	9	7	7	16	16	12	6	6	6	5	7	7	7	3	2

Şekil 3. Örnek problem için kaynak seviyeleme öncesi çubuk diyagramı ve günlük kaynak kullanım miktarları

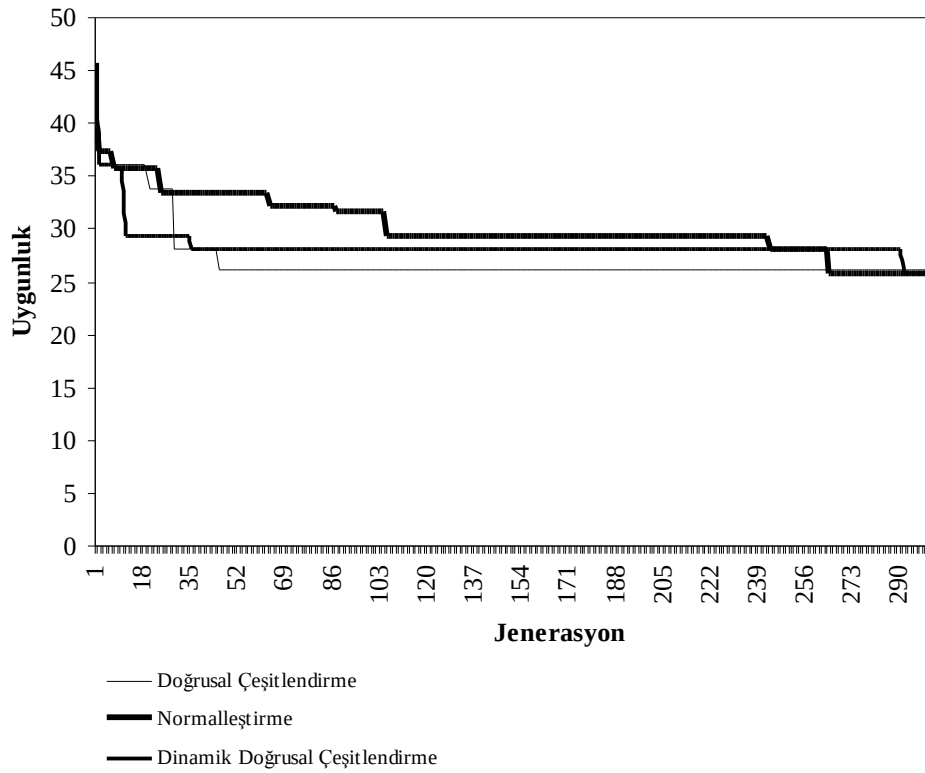
Program					Zaman															
Akt	Kay	Baş	Bitiş	Süre	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	2	0	2	2	■	■														
B	1	0	4	4	■	■	■													
C	4	0	1	1	■															
D	4	2	6	4										■	■	■	■	■		
E	2	4	7	3								■	■	■						
F	4	4	10	6					■	■	■	■	■	■						
G	6	1	7	6		■	■	■	■	■	■									
H	0	7	8	1															■	
J	2	7	11	4											■	■	■	■	■	
K	1	10	15	5											■	■	■	■	■	
L	2	15	16	1																■
Günlük kaynak kullanımı					7	9	7	7	10	10	10	6	6	10	5	7	7	7	3	2

Şekil 4. Örnek problem için kaynak seviyeleme sonrası çubuk diyagramı ve günlük kaynak kullanım miktarları

Projenin hesaplanan OKD değeri 6.8'dir. Seviyeleme yapılmadan önceki z değeri 69, programın seviyeleme yapması sonucunda bulunan z değeri ise 25.75'dir. Bu değer Leu ve ark. (1) tarafından da 25 olarak elde edilmiştir.

Programda kullanılan parametre değerleri (Tablo3) sabit tutulup çeşitlendirme yöntemlerinin verimliliği denendiğinde normalleştirme' nin sonuca en çabuk ulaştırdığı görülmektedir. Parametre değerleri değiştirildiğinde de normalleştirme en iyi sonucu vermiştir.

Şekil 5 incelendiğinde en iyi uygunluk değeri $z=25.750$ Normalleştirme sonucunda 264. kuşakta elde edilmektedir. Dinamik Doğrusal Çeşitlendirme Yönteminde ise bu değere 291. kuşakta ulaşılmıştır. Doğrusal Çeşitlendirme Yönteminde ise 300 kuşak sonunda dahi bu değere ulaşamamıştır.



Şekil 5. Çeşitlendirme tiplerine göre değişen uygunluk değeri/ jenerasyon sayısı

5. SONUÇ VE ÖNERİLER

Genetik Algoritma optimizasyon problem çözümüne yaklaşımı itibarı ile kaynak seviyeleme problem çözümleri için çok uygundur. Literatürde konu ile ilgili az sayıda da olsa çalışma bulunmaktadır. Bu çalışma kapsamında, bir bilgisayar yazılımı geliştirilmiş ve validasyonu yapılmıştır. Bu yazılımda, literatürde tartışılan programlara ek olarak üç tip çeşitlendirme yöntemi denenmiş, normalleştirmenin en hızlı sonuçları verdiği gözlenmiştir.

Bu çalışma sırasında tek tip kaynak seviyelemesi üzerine yoğunlaşmıştır. Birden fazla kaynağın seviyelemesi ve kaynak seviyeleme ile birlikte zaman/maliyet optimizasyonu ileriki çalışmalar için önerilmektedir.

6. KAYNAKLAR

1. Leu,S., Yang,C. and Huang,J., “Resource Leveling in Construction by Genetic Algorithm-Based Optimization and Its Decision Support System Application”, *Automation in Construction*, 10, 27-41, 2000.
2. Ogbu,O.O., Tah,J.H.M., “Development and Application of a Hybrid Genetic Algorithm for Resource Optimization and Management”, *Engineering, Construction and Architectural Management*, 4, 307-317, 2002.
3. Gen,M., Cheng,R., “Genetic Algorithms And Engineering Design”, (H.R. Parsei, Editor) John Wiley&Sons Inc., 96-8437, 411, United States of America, 1997.
4. Oğuz,M. ve Akbaş,S., “Genetik Algoritmalar”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, 41s., İstanbul, 2000.
5. Easa,S.M., “Resouce Levelling in Construcction by Optimization”, *Journal of Construction Engineering and Management*, ASCE 115(2), 302-316, 1989.
6. Ahuja,H., “Construction Performance Control By Networks”, Wiley, New York, 1976.
7. Harris,R.B., “Precedence and Arrow Networking Techniques for Construction”, Wiley, New York, 1978.
8. Harris,R..B., “Packing Method for Resource Leveling (PACK)”, *Journal of Construction Engineering and Management*, ASCE 116(2), 331-350, 1990.
9. Hegazy,T., “Optimization of Resource Allocation and Levelling Using Genetic Algorithm”, *Journal of Construction Engineering and Management*, 125 (3), 167-175, 1999.
10. Al-Tabtabai,H., Alex,A.P., “Using Genetic Algorithms to Solve Optimization Problems in Construction”, *Engineering, Construction and Architectural Management*, 6 (2), 121-132, 1999.