

# Doğrusal Genetik Programlama Algoritmasının Paralel Mimariye Uyarlanması İçin Yeni Bir Model Önerimi

## A New Model Proposal for Adapting the Linear Genetic Programming Algorithm to Parallel Architecture

Özgür AKSU<sup>1</sup>, Adem KALINLI<sup>2</sup>

1. Bilgisayar Mühendisliği Bölümü  
Erciyes Üniversitesi  
oaksu@erciyes.edu.tr

2. Kayseri Meslek Yüksekokulu  
Erciyes Üniversitesi  
kalinlia@erciyes.edu.tr

### Özet

Birçok mühendislik problemin çözümü matematiksel modelleme ve benzetim yaklaşımları ile çözümlenmesine rağmen tam olarak çözülemeyen çok sayıda zor mühendislik problemi bulunmaktadır. Zor mühendislik problemlerinin çözümü için araştırmacıların ilgisi giderek artan oranda yapay zekâ metotlarının geliştirilmesine ve kullanılmasına yönelmiş bulunmaktadır. Literatürdeki yapay zekâ metotlarından birisi topoloji optimizasyonu da içeren Doğrusal Genetik Programlama (LGP) algoritmasıdır. LGP temelde tek çekirdekli bilgisayar sistemi için tasarlanmıştır. Bu çalışmada, LGP algoritmasının OpenMPI Kütüphanesi kullanılarak paralelleştirilmesine dayalı yeni bir paralel LGP modeli önerilmiş ve önerilen modelin performansı diğer bazı yöntemlerle karşılaştırılmıştır.

### Abstract

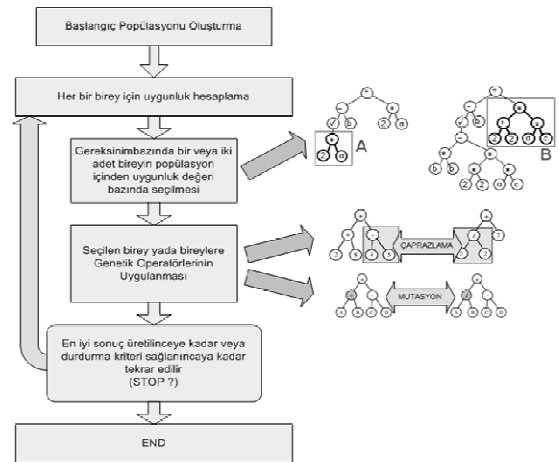
Many engineering solution of the problem solved by mathematical modeling and simulation approaches can not be solved exactly, although there are many difficult engineering problems. For the solution of difficult engineering problems to the attention of researchers increasingly turning to artificial intelligence methods has been developed and used. Artificial intelligence methods in the literature, including one of the topology optimization Linear Genetic Programming (LGP) algorithm. LGP is basically designed for a single-core computer systems. In this study, using Library LGP OpenMP parallelized algorithm based on a new model is proposed LGP parallel performance of the model are compared with some other methods.

### 1. Giriş

Geleneksel algoritmalar genellikle küçük boyutlu problemlerin çözümünde iyi sonuçlar vermektedir. Bu nedenle, tasarım, üretim, kontrol ve simülasyon gibi pratik alanlardaki optimizasyon problemlerinin büyük zorluğu, araştırmacıları yeni ve güçlü algoritmalar geliştirmeye teşvik etmiştir. Bu amaçla araştırmacıların ilgisi yapay zekâ ve tabii bilimler üzerine yoğunlaşmış ve geliştirilen çeşitli algoritmalar bir çok zor problemin çözümünde başarılı sonuçlar sağlamıştır.

Bu süreç içinde geliştirilen yapay zekâ algoritmalarından biri olan Genetik Algoritmalar (GA), yapay zekanın önemli bir dalıdır. GA Darwin'in evrim teorisinden esinlenerek

oluşturulan evrimsel hesaplama (evolutionary computing) tekniğinin bir parçasıdır. Evrimsel hesaplama, ilk olarak 1960'larda I. Rechenberg tarafından "Evrimsel Stratejileri" adlı eserinde gündeme getirilmiştir [1]. Literatürde, evrimsel hesaplama sınıfında yer alan çok sayıda algoritma ve tekniğin yanı sıra bu yöntemlerin çeşitli özelliklerini bir arada kullanan çok sayıda melez sistemler de mevcuttur. Ancak genel olarak bu guruba giren temel algoritmalar örnek olarak GA, Evrimsel Programlama (EP), Evrimsel Stratejiler (ES) ve Genetik Programlama (GP) teknikleri sıralanabilir. Parametre optimizasyonu olarak temel alınan yapay zekâ yöntemlerine alternatif olarak 1985'de Michael L. Cramer tarafından geliştirilen ve John Koza tarafından uygulanabilir hale getirilerek zor mühendislik problemlerinden biri olan Elektronik Devre Modeli Üretimi için uygulanan GP algoritması çözüme, eşdeğer model üretmek katkıda bulunmaktadır [2,3]. GP Algoritması, diğer yapay zekâ algoritmalarından, sonuca ulaşmak için bir yapısal eşitlik veya yaklaşım gerektirmeden çalışabilmesi ile kesin çizgilerle ayrılmaktadır. Eğer problem bir formülasyon ile tanımlanamıyor, çözümün bir tanımı başta yapılamıyorsa problemi çözebilecek en etkili yöntemlerden birisi, topoloji araştırma algoritması olan GP algoritmasıdır. Temel bir GP algoritması Şekil 1'de yer alan adımlardan oluşmaktadır. GP Algoritma modelinde hedef amaç için çözüm adayları eleman kümesi kullanılarak hedef amaca en uygun çözüm bireyinin geliştirilmesi amaçlanmaktadır.

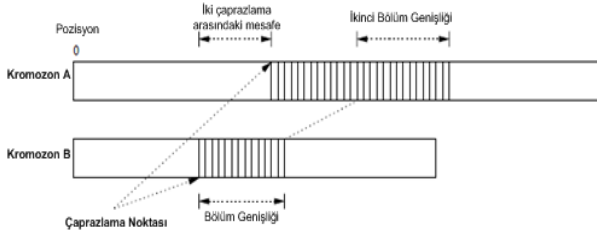


Şekil 1: GP Algoritması Modeli.

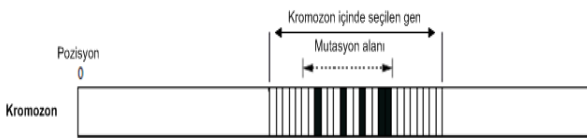
Sunulan bu çalışmada performans ve çalışma verimliliği açısından diğer GP türlerinden öne çıkan LGP Algoritması temel alınmaktadır. Çalışmada öncelikle temel açıdan LGP Algoritması incelenmiş ardından ilgili modelin etkin bir şekilde paralelleştirilmesi için gerekli olan model içi operatör yaklaşımlarına yer verilmektedir. Elde edilen gelişimin karşılaştırılması için benzer çözümlerle karşılaştırmalar yapılarak sunulmaktadır.

## 2. Doğrusal Genetik Programlama ve Parallelleştirilmesi

Araştırmacılar GP geliştirildiği 1985 yılından günümüze, algoritmanın var olan performans ve adaptasyonunu geliştirmek için gayret göstermektedirler. Süreç içerisinde bilgisayar yazılım ve donanım gereksinimleri bazında yeni bir model ortaya konmuştur; LGP. LGP Modeli ilk olarak 1997 yılında Ağaç Gösterimli GP (TGP) yerine önerilmiştir. Yazılım geliştirme süreci açısından LGP bireyleri tanımlanması ve işletilmesi çok daha pratik olan tek boyutlu dizler üzerinde tutulmaktadır. LGP Algoritmasında birey kodlama ve çözümlenme açısından yazılımsal işlem verimliliği sağlanmıştır. Tek boyutlu dizi üzerinde tutulan bireyler üzerinde GP algoritma adımları işletilmektedir. Temel işlem adımlarının aynı olmasına karşın kromozom saklama ve kullanım şekllinden ötürü genetik operatörlerin uygulanmasında farklılıklar ortaya çıkmıştır. Şekil 2’de Çaprazlama ve Şekil 3’de Mutasyon operatörlerinin uygulanmasına yönelik örnek yaklaşımlar gösterilmektedir [4-6].



Şekil 2: LGP’da çaprazlama operatörünün uygulanması.



Şekil 3: LGP’da mutasyon operatörünün uygulanması.

### 2.1 Önerilen Model ile LGP’nin Parallelleştirilmesi

LGP Algoritması çözümü zor olan mühendislik problemlerin çözümünü geliştirmekte başarı gösterse de problem boyutuna bağlı çözüm uzayı büyüklüğü temelinde hesaplama zamanı sorunu yaşamaktadır. Problemlerin zorluk derecesi ile paralel olarak büyüyen çözüm uzayı problem çözüm sürecini de arttırmaktadır. 2011 Yılında literatürde yeni bir LGP yaklaşımı olarak PLGP (Paralel Lineer Genetic Programming) önerilmektedir. PLGP Modelinde var olan çaprazlama ve mutasyon operatör işlemleri alt birimlere dağıtılarak işlem yükü azaltılarak performansın artırılması sağlanmaktadır [10].

Yapılan algoritma testleri sonucunda PLGP modelinin düşük seviyede yakınsamaya sahip olduğu ortaya çıkmıştır. Daha yüksek yakınsama ve performans başarımları için modelin operatör bazında kalmadan paralelleştirilmesi önerilmektedir [11]. Buna ek olarak paralel modellerin ihtiyaç duyduğu operatörlerinde modele dahil edilmesi gerekliliği ortaya çıkmaktadır [12].

Hazırlanan bu çalışmada önerilen çözüm modelinde LGP algoritması operatörleri ve tasarımı gözetilerek paralel donanım gereksinimleri dikkate alınarak araştırmalar yapılmıştır. Yapılan araştırmalar sonucunda işlem yükü artan bir algoritmanın daha yüksek işlem zamanı kapasitesi için paralelleştirilmesinin var olan performansı arttıracığı tespit edilmiştir. NUMA Mimarisi temel alınarak seçilmiş olan paralel programlama modeli Paralel Alt Uzay arama modelidir [9]. Paralel alt uzay arama metodu ile paralelleştirilmek istenen LGP algoritması temelinde sabit uzunluklu olmayan başlangıç tanımlı tek boyutlu dizlerin kromozom olarak kullanılması önerilmektedir. Her bir kromozom geni bir karakter tanımı ile başlamaktadır. İlgili tek karakter tanımlaması özetli Çizelge 1’de verilmektedir. Önerilen yapı alfasayısal tek boyutlu dizler üzerinden tutulabilmekte ve transferi yapılabilmektedir [4].

Çizelge 1: LGP Birey içinde yer alan kodlama tanımları.

İlk Karakter Tanımı	İçeriği	Yapısal Şekli	Örnek Gösterim
x	Değişken tanımı (3 byte)	x, işlem numarası[1 byte], değişken numarası[1 byte]	x,#1,#0 Reel karşılığı x-
?	Sabit sayı tanımı (10 byte)	?, işlem numarası[1 byte], Double sayı tanımı [8 byte]	?,#0,#0#0#0#0#0 #0#240#63 Reel karşılığı 1+
y	İkincil işlem içeren değişken tanımı (4 byte)	y, işlem numarası[1 byte], ikinci işlem numarası[1 byte], değişken numarası[1 byte]	y,#2,#0,#0 Reel karşılığı sin(x)*
@	İkincil işlem içeren sabit sayı tanımı (11 byte)	@, işlem numarası[1 byte], ikinci işlem numarası[1 byte], Double sayı tanımı [8 byte]	@,#3,#1, #0#0#0#0#0#0#2 40#191 Reel karşılığı cos(-1)/

LGP Algoritması tasarımda yer alan tek boyutlu dizi tanımının alfasayısal değişken tanımı ile paralel alt uzay araştırma işlemi için OpenMPI Kütüphanesi kullanılmıştır. OpenMPI Kütüphanesi dinamik ve esnek iletişim desteği ile fonksiyonlarına uygunlaştırılması sonucunda donanımsal ve yazılımsal olarak yapay zekâ algoritmasının gerçekleştirilmesi sağlanmaktadır.

Geliştirilen yaklaşımın aynı anda birden çok terminal üzerinde bağımsız olarak çalıştırılması gerekliliği kapsamında aşağıdaki sorunlar ve ilgili sorunlar için OpenMPI Kütüphanesi bazında geliştirilen çözüm yaklaşımları listelenmiştir:

- *Benzersiz bireylerin oluşturulması:* Her bir terminalde ayrı ayrı başlayan LGP algoritma sürecinde başlangıç popülasyonu bireylerinin diğer terminallerde kullanılan bireylerden farklı üretilmesi ve benzersiz bireyler ile algoritmanın başlatılması hedeflenmiştir. Bu amaçla terminallerden biri popülasyon takip ve üretim amacı ile kullanılmıştır. Her bir alt terminal birey talebinde

bulunurken bu terminalden birey talep etmektedir. Benzersiz üretim hedefi doğrultusunda üretim terminali üretilen tüm bireyleri geçmiş matrisinde tutmakta ve yeni üretimlerde eski üretimleri dikkate alarak gerçekleştirmektedir. Geliştirilen ve LGP algoritmasına adapte edilen bu yaklaşımla algoritma gelişimindeki tekrarların, gereksiz işlem zamanlarının önüne geçilmesi amaçlanmıştır:

- **Çözüm geçmişi:** Terminallerde sürdürülen algoritma gelişim sürecinde elde edilen çözümler terminal içinde yerel hafıza üzerinde depolanmaktadır. Aynı zamanda terminaler üzerinde tutulan çözümler aynı zamanda ana üretim terminaline iletilmektedir. İletilen çözüm gelişim bilgileri sayesinde terminallerin gelişimi izlenebilmekte ve olası yerel minimum döngüsel boşluğuna engel olunmaktadır. Çözümün gelişiminin terminal bazında izlenmesi süreci ağ kaynağını olumsuz yönde etkilemede algoritmanın yeni çözüme, hedef amaca ulaşmasının takibinde önemli bir bilgi kaynağını oluşturmaktadır.
- **En iyi çözümün korunması:** Çözüm geçmişi bilgisinin üretim terminalinde tutulması işleminde terminallere ait en iyi çözümlerin ve algoritma genelinin en iyi çözümünde takip edilmesi sağlanmaktadır. Üretim terminalinin yeni bireyler üretirken var olan en iyi çözüm ve çözümü oluşturan birey tanımını dikkate alarak diğer terminalleri de hedef çözüm alanına yönlendirmesi mümkün olmaktadır.
- **Değişken kontrol parametreleri:** Üretim terminali paralel çalışan terminallere birey üretip, terminallerin elde ettiği çözümleri takip ederken aynı zamanda yeni çözüm gelişimini takip etmektedir. Probleme ve problemin gelişim adımları sürecinde oluşacak gereksiz işlem ve zaman kayıplarını minimuma indirmek için terminallerde işletilen algoritma parametrelerini değiştirmektedir.
- **Yakın Atama:** Çözüm gelişim sürecinde terminallerin gelişiminde yaşanacak kendini tekrarlama ve yeni çözüm geliştirememesi gibi sorunlar tespit edildiğinde var olan en iyi çözüm temel alınarak terminal popülasyonu yakın atama operatörü ile yeniden oluşturulmaktadır. Böylece terminal kendi kısır döngüsünden çıkarak mevcut en iyi çözümün etrafında araştırma yapmakta, olası daha iyi bir çözümü araştırabilmektedir.
- **Katkılama:** Yakın atama operatörü kullanılmasına rağmen belirli bir çevrim sürecinde gelişim sağlayamayan terminal popülasyonu tamamen daha önce hiç araştırılmamış bireylerden yeniden oluşturmaktadır. Katkılama olarak tanımlanan bu süreçte terminal çözüm arayışında hiç bakılmamış çözüm uzayı bölümlerine yönlendirilmektedir.

Var olan LGP algoritmasının aynı anda birden çok paralel terminalde yürütülmesinde tespit edilen ve çözüm önerisi getirilen sorunları OpenMPI Kütüphanesi ile yeniden modellenmiş ve kodlanarak gerçekleştirimi sağlanmıştır. Şekil 4’de önerilen ve paralel donanıma adapte edilen Paralleleştirilmiş LGP, GENAN modeli verilmektedir. Verilen model içinde paralel sistem erişim adımları dairesel döngü tanımları ile gösterilmektedir.

### 3. Önerilen Paralel Modelin Test Edilmesi

Önerilen GENAN modelinin başarımının ve performansının daha önce incelemesi yapılmış diğer GP algoritmaları ile

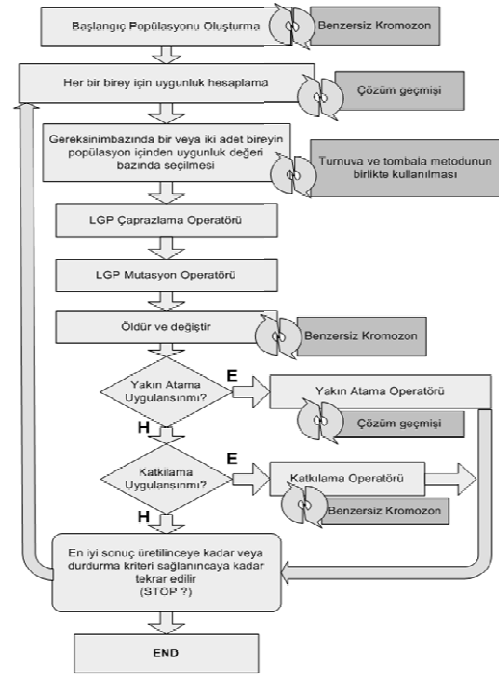
karşılaştırılabilmesi için Çizelge 2’de verilen sürekli test fonksiyonları kullanılmıştır [5-8].

Test süreci University Alabama Birmingham (UAB) bünyesindeki Cheaha yüksek başarılı hesaplama merkezinde gerçekleştirilmiştir. Test sırasında 256 adet işlemci kullanılmıştır. Terminaler 2.66Ghz işlemci hızına 4Gb hafızaya sahiptir.

Algoritma başarımını karşılaştırmak amacıyla 100 farklı koşma sonucunun karşılaştırılması Şekil 5-7’de verilmektedir. İlgili grafiklerde kullanılan MEP (Multi-Expression Programming) algoritması, GEP (Gene Expression Programming) algoritması, GE (Grammatical Evolution) algoritması, LGP Doğrusal Genetik Programlama algoritması ve GENAN önerilen model algoritmasını simgelemektedir. Test sürecinde referans alınan karşılaştırma sonuçları için geliştirilen ve önerilen GENAN modeline ait diğer parametre ve tanımlamalar Çizelge 3’de verilmektedir.

Şekil 5-7’de verilen Başarı Oranı (Success rate) hesaplaması için Eşitlik 1’de verilen fonksiyon kullanılmıştır [7]. Elde edilen sonuçların genel özeti Çizelge 4’de verilmektedir.

$$\text{Başarı Oranı} = \frac{\text{Başarılı Koşmaların Sayısı}}{\text{Toplam Koşma Sayısı}} \quad (1)$$



Şekil 4: Önerilen LGP Algoritma (GENAN) Modeli.

Çizelge 2: Performans karşılaştırma amaçlı seçilen LGP sürekli test fonksiyonları [5-8].

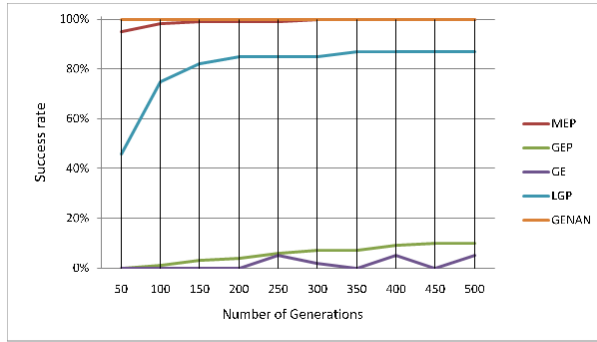
Fonksiyon Eşitliği	Fonksiyon Parametreleri
$f_1(x) = x^4 + x^3 + x^2 + x$	Popülasyon genişliği 50 kromozom, F={+, -, *, /, Sin}
$f_2(x) = x^6 - 2x^4 + x^2$	Popülasyon genişliği 50 kromozom, F={+, -, *, /, Sin}
$f_3(x) = \text{Sin}(x^4 + x^2)$	Popülasyon genişliği 50 kromozom, F={+, -, *, /, Sin}

Çizelge 3: Kullanılan algoritma parametreleri.

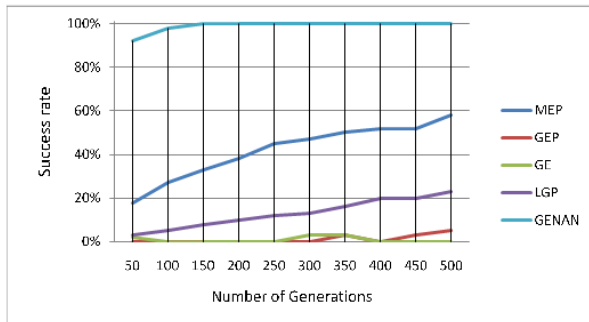
Parametre	Verilen Değer
Çaprazlama oranı	%90/%50 (in:%30/%50)
Mutasyon oranı	%10/%50 (in:%10/%50)
Yakın atama eşik değeri	%10
Maksimum iterasyon	500
Maksimum parametre sayısı	7 gen
İşlem kümesi elemanları	-, +, *, /, Sin
Uygunluk Fonksiyonu (u(x) üretilen çözüm adayını temsil etmektedir)	$\sum_{x=0}^{\text{Max}(x)} \frac{1}{1 + \text{Abs}(f(x) - u(x))}$

Çizelge 4: Test fonksiyonları sonuçları [7,13].

Test Fonksiyonu	MEP	GEP	GE	LGP	GENAN
F1	99%	6%	2%	81%	100%
F2	40%	1%	1%	13%	90%
F3	49%	7%	11%	53%	69%



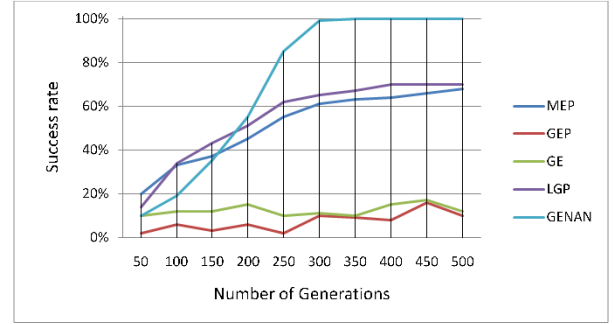
Şekil 5: F1 Fonksiyonu test sonuçlarının diğer algoritmalar ile karşılaştırılması [7,13].



Şekil 6: F2 Fonksiyonu test sonuçlarının diğer algoritmalar ile karşılaştırılması [7,13].

#### 4. Sonuçlar

Bu çalışmada, MPI yapısına dayalı olarak LGP algoritmasının paralelleştirilmesine yönelik bir yaklaşım önerilmiştir. Önerilen yaklaşımın performansını sürekli test fonksiyonları üzerinde incelenmiş ve diğer bazı yöntemlerin başarımı ile karşılaştırılmıştır. Elde edilen sonuçlardan, birim zamanda gerçekleştirilen algoritma adım sayısının artmasının ve birden çok terminal üzerinden paralel araştırma gerçekleştirilmesinin başarımı olumlu yönde etkilediği görülmüştür. Bu kapsamda GENAN olarak adlandırılan yaklaşımın performansının



Şekil 7: F3 Fonksiyonu test sonuçlarının diğer algoritmalar ile karşılaştırılması [7,13].

dikkate alınan diğer yöntemlere göre anlamlı düzeyde yüksek olduğu görülmüştür. Çalışma sonrasında elde edilen algoritma başarımının daha da artırılması ve büyük ölçekli problemlerin çözümünde geliştirilen yaklaşımın kullanılabilmesi için algoritma parametrelerinin paralelleştirme sonrası gereksinimler gözetilerek adapte edilmesi, birleşik parametre optimizasyonunun yapılmasının sonucunda var olan performansın artışına katkıda bulunacağı düşünülmektedir.

#### 5. Kaynaklar

- [1] Rechenberg, I., Evolutionsstrategie, Frommann-Holzboog, Stuttgart, Germany, 1973.
- [2] N.L. Cramer, A Representation for the Adaptive Generation of Simple Sequential Programs, Proceedings Of An International Conference On Genetic Algorithms and Their Applications, pp. 183-187, 1985.
- [3] J.R. Koza, Genetic Programming: On the Programming of Computer Programs by Natural Selection. MIT Press, Cambridge, MA, 1992.
- [4] M. Brameier, W. Banzhaf, Linear Genetic Programming, Springer, ISBN 0-387-31029-0, 2007.
- [5] P.J. Nordin, Evolutionary Program Induction of Binary Machine Code and its Applications. PhD thesis, University of Dortmund, 1997.
- [6] W. Banzhaf, Genetic Programming for Pedestrians. In S. Forrest (ed.) Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93), p. 628, Morgan Kaufmann, San Francisco, CA, 1993.
- [7] M. Oltean, C. Groşan, A Comparison of Several Linear Genetic Programming Techniques, Complex Systems Publications, Vol. 14, pp. 285-313, 2003.
- [8] W. Fan, E. A. Fox, The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery For Web Search, Journal of the American Society for Information Science and Technology, Vol. 55, 2004.
- [9] Pacheco, P. S., An Introduction To Parallel Programming, Morgan Kaufmann, 2012.
- [10] C. Downey and M. Zhang. Parallel linear genetic programming. In EuroGP, vol. 6621, pp. 178-189, 2011.
- [11] Downey, C., Explorations in Parallel Linear Genetic Programming. Master's thesis, Victoria University of Wellington, New Zealand, 2011.
- [12] Scoble, A., Johnston, M., Zhang, M., Local Search in Parallel Linear Genetic Programming for Multiclass Classification, AI 2012, LNCS 7691, pp. 373-384, 2012.
- [13] Aksu, Ö., Kalınlı, A., Doğrusal Genetik Programlama Algoritmasının Dağıtık Hesaplama Mimarisine Uyarlanması, 3. Ulusal Yüksek Başarımlı Hesaplama Konferansı 12-13 Nisan 2012.