

Genetik Algoritmalarda ‘Injection’ Yaklaşımı ile Genetik Çeşitliliğin Korunması

Mustafa ORAL¹

Serkan KARTAL²

^{1,2}Çukurova Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü

¹moral@cu.edu.tr

²skartal@cu.edu.tr

Özet

Günümüzde, Genetik Algoritma'nın (GA) farklı versiyonları bir çok optimizasyon probleminin çözümünde kullanılmaktadır. Ancak, zamanla genetik çeşitliliğin kaybedilmesi ve zamanından önce yerel en iyi çözüme yakınsanması, halen GA'nın en temel problemlerinden birisini oluşturmaktadır. Bu çalışmanın amacı, GA'nın çalışması süresince belirli aralıklarla, popülasyona farklı yöntemlerle elde edilmiş yeni bireylerin eklenmesinin genetik çeşitliliğe etkilerini gözlemlemektir. Böylece, algoritmanın yerel en iyi çözüme takılmasının önüne geçilerek, genel en iyi çözüme daha kolay bir şekilde ulaşması hedeflenmektedir. Öne sürülen algoritmalar, içerisinde yerel çözümlerin de bulunduğu genel optimizasyon test fonksiyonları ile test edilmiş, elde edilen sonuçlar standart GA'nın sonuçları ile karşılaştırılmıştır. Sunulan algoritmaların kaybedilen genetik çeşitliliği yeniden arttırdığı, bunda genel en iyi çözümü bulmayı kolaylaştırıp, GA'nın performansında iyileşme sağladığı gözlemlenmiştir.

1. Giriş

Genetik algoritma, doğal seçim mekanizmasından esinlenerek kurulmuş, genetik tabanlı sezgisel bir yaklaşımdır. 1970 yılında John HOLLAND'ın yapmış olduğu çalışma ile popülerlik kazanmıştır[1]. Evrimsel algoritmaların bir alt başlığı olup genelde optimizasyon problemlerinin çözümünde kullanılmakta, canlıların doğal evrim sürecindeki aşamalarını taklit ederek elde edilen çözümlerden daha iyi çözümler elde etmeyi amaçlamaktadır. GA'lar da olası her çözüm bir birey ile temsil edilmekte ve bu bireylerin oluşturduğu topluluk popülasyon olarak adlandırılmaktadır. Popülasyon bir kere oluşturulduktan sonra, belirli bir uygunlukta çözüm bulunana kadar veya belirli bir jenerasyon boyunca seçim, çaprazlama ve yeni popülasyonunun oluşturulması işlemleri tekrarlanmaktadır.

GA'nın performansını etkileyen en önemli iki faktör sırası ile başlangıç popülasyonunun oluşturulması ve genetik çeşitliliğin jenerasyonlar boyunca korunmasıdır[2,3]. Başlangıç

popülasyonunun iyi oluşturulmadığı, yani arama uzayının her bölümünü temsil eden bireylerin bulunmadığı durumlarda, GA'nın iyi bir sonuç üretme olasılığı da düşük olacaktır. Özellikle, genel en iyi çözümün, başlangıç popülasyonundaki bireylerden uzak bir yerde olduğu durumlarda, genel en iyi çözüme ulaşılması şansı sadece mutasyon işlemi ile mümkün olabilmektedir. Bununla birlikte, başlangıç popülasyonun iyi bir şekilde oluşturulabildiği durumlarda dahi, seçim işlemine bağlı olarak jenerasyonlar boyunca popülasyondaki tüm bireyler hızlı bir şekilde en iyi bireylere benzemeye başlayabilir, yani genetik çeşitlilik hızla kaybedilebilir. Bunun sonucunda da, özellikle içerisinde yerel en iyi çözümleri barındıran problemlerde, GA, genel en iyi çözüm yerine hızlı bir şekilde yerel en iyi çözüme yönelebilir. Bu durum sonucunda ise genel en iyi çözüme hiçbir zaman ulaşamayabilir.

Bu çalışmada, hem GA'nın performansının başlangıç popülasyonuna olan bağımlılığını azaltma, hem de jenerasyonlar boyunca genetik çeşitliliği korumak amacıyla, "Injection" yaklaşımı kullanılmıştır. Böylece, algoritmanın genel en iyi çözüme ulaşma şansı artırılmaya çalışılmıştır. Injection metodu ile, her jenerasyon sonunda, popülasyonda bulunan belirli bir oranda birey yok edilerek, yerine rastgele oluşturulmuş yeni bireyler popülasyona eklenmektedir. Böylece, hem var olan bireylerle daha iyiye doğru evrimleşme işlemi gerçekleşirken, hem de yeni bireylerin popülasyona eklenmesi ile genetik çeşitlilik korunmuş olmaktadır.

2. Genetik Çeşitliliğin Korunması

Genetik çeşitliliğin azalması, yerel iyi çözümleri olan fonksiyonlarda, algoritmanın kısa bir sürede yerel iyi çözüme yakınsamasına neden olup, genel en iyi çözümü bulamadan sonlanma olasılığını arttırmaktadır[4]. Çeşitlik koruma metodları farklı genetik özellikte bireylerin popülasyonda bulunma şansını artırarak problem alanının daha geniş bir yüzeyinin aranmasını sağlamakta ve yerel çözüme yakınsama probleminin ortaya çıkma ihtimalini azaltmaktadır.

Genetik çeşitliliğin korunması için öne sürülen yaklaşımlardan birisi “crowding” tir. Bu yaklaşıma göre ilk olarak popülasyon alt popülasyonlara ayrılmakta ve oluşturulan yeni bireyler bu alt popülasyona eklenmektedir. Eklenme sırasında alt popülasyonda bulunan ve yeni eklenen bireye genetik olarak en çok benzeyen birey çıkarılmaktadır. Alt popülasyonun büyüklüğü “crowding factor” parametresi ile belirlenmektedir [5].

Shimodaira “Diversity Control Oriented Genetic Algorithm (DCGA)” metodu ile popülasyon içerisindeki çeşitliliği korumaya çalışmıştır[6]. Bu algoritmada, bir sonraki jenerasyon için popülasyon oluşturulurken, çocukların oluşturduğu yeni popülasyon ile anne-babaların oluşturduğu eski popülasyon harmanlanmaktadır. Bu işlem sırasında, her iki popülasyon birleştirilmekte, ancak birbirine benzer olan bireyler yok edilerek, yalnızca farklı bireylerin oluşturduğu yeni bir popülasyon elde edilmektedir. Bireylerin benzerlikleri Hamming mesafesi ile ölçülmektedir. Yine benzer bir uygulama Mauldin tarafından Hamming mesafesi yardımıyla, benzer özellikteki bireylerin elenmesi işlemi şeklinde gerçekleştirilmiştir [7].

Öne sürülen bir başka algoritma ise “Ranked Space” algoritmasıdır[8]. Bu algoritma genetik çeşitliliği seçim işlemi sırasında bireylerin seçim şansına etki ederek korumaktadır. Bu algoritma ile seçim işlemi sırasında iki farklı sıra puanı (rank score) hesaplanmaktadır. Bunlardan birincisi uygunluk (fitness) değerine bağlı olarak verilen sıra puanı, ikincisi ise genetik farklılığa bağlı olarak verilen sıra puanıdır. Her iki puanın toplamı ise bireylerin seçim puanlarını oluşturmaktadır. Böylece genetik farklılığı yüksek olan bireylerin de seçim şansını artırılarak, genetik çeşitlilik korunmaya çalışılmaktadır.

Benzer bir yaklaşım olan “Sharing” metodu ile aynı genetik yapıya sahip olan bireylerin seçim şansları benzer birey sayısına bölünmektedir. Bu işlem sonunda benzer bireylerin seçim şansları azaltılarak tüm popülasyonun benzer bireylerden oluşmasının önüne geçilmektedir [9]. Böylece farklı genetik özelliklerdeki bireylerin seçim şansını artırmakta ve genetik çeşitlilik kısmen korunabilmektedir.

Yukarıdaki çalışmalardan farklı olarak bu çalışmada genetik çeşitliliği korumak ve yerel çözümlere takılmadan genel en iyi çözüme ulaşabilmek için “Injection” yaklaşımı kullanılmıştır. Bu yaklaşım ile, genetik çeşitliliğin korunması yalnızca popülasyon içerisindeki bireylere bağlı olmamakta, aynı zamanda daha önce popülasyonda hiç bulunmayan bireylerinde popülasyona eklenmesi ile çözüm tüm problem uzayında aranabilmektedir.

3. Yöntem

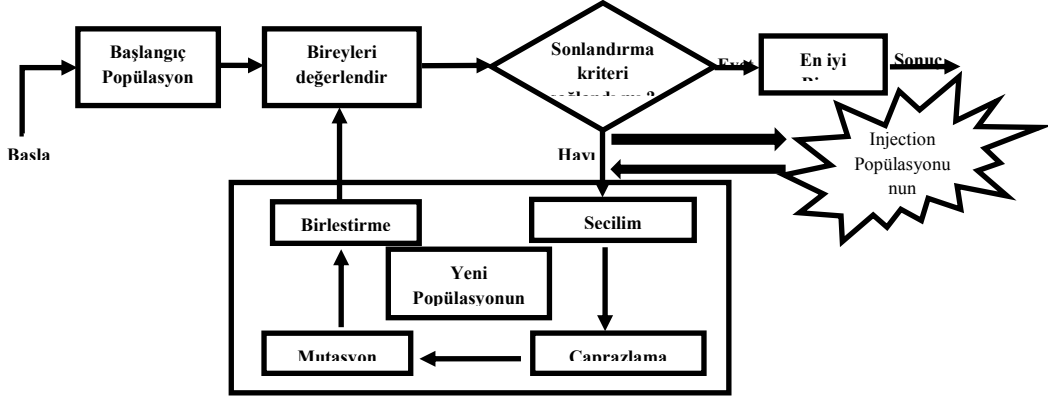
Standart Genetik Algoritma (SGA), Darwin’in evrim teorisinden esinlenerek kurulmuş, problem uzayını arayarak en iyi çözümü bulmaya çalışan bir algoritmadır [10]. Genelde belirli bir sayıda jenerasyon oluşturulduktan sonra işlem sonlandırılır. GA’nın genel yapısı Şekil 1’de verilmiştir.

SGA çalışmaya rastgele oluşturulan ve başlangıç popülasyonu olarak adlandırılan bireyler (çözüm adayları) ile başlar. Popülasyon büyüklüğü sabit bir değer olmakla birlikte, problemin büyüklüğüne göre belirlenmektedir. Algoritma, iyi bireylerin hayatta kalması ilkesine bağlı olarak popülasyonun jenerasyonlar boyunca daha iyi çözümlere doğru evrimleşmesini sağlar. Her jenerasyonda amaç fonksiyonu ile bireylerin probleme olan uygunluk değerleri hesaplanır. Eğer istenen uygunlukta bir çözüm bulunmuş ise veya belirlenen sayıda jenerasyon gerçekleşmiş ise algoritmanın çalışması sonlandırılır. İstenen koşullar oluşmuyorsa bir sonraki jenerasyon oluşturulmalıdır.

Yeni jenerasyonun oluşturulması seçim işlemi ile başlar. Bu aşamada turnuva ve rulet tekeri gibi farklı seçim metodları kullanılabilir. Bir sonraki adımda, seçilen bireyler yeni bireyleri oluşmak üzere eşleştirilir. Eşleşme sonucunda yeni bireylerin oluşup oluşmayacağına çaprazlama olasılığı olan p_c (crossingover probability) değeri ile karar verilir. Yeni bireylerin oluşmaması durumunda, anne ve babaların birer kopyası direk olarak yeni popülasyona eklenir. Yeni oluşturulan bireyler, popülasyona eklenmeden önce belirli bir olasılık değerine göre mutasyon işlemine tabi tutulurlar. Mutasyonun amacı, çok düşük bir olasılık ile bireylerin genlerinde rastgele değişiklik yaparak genetik çeşitliliğe katkıda bulunmaktır. Yeni bireylerin oluşturulma işlemi tamamlandıktan sonra, eski popülasyon ile yeni popülasyonun birleştirilmesi ile bir sonraki jenerasyonun oluşturulur. Yeni jenerasyon, eski popülasyondan elit birey oranı kadar en iyi bireylerin alınması ve geri kalan yerler içinde yeni popülasyondaki bireylerin alınması ile oluşturulur. Tüm bu işlemler sonlanma koşulu sağlanana kadar devam eder.

Bu çalışmada SGA ya ek olarak her jenerasyonda seçim işlemine geçilmeden önce popülasyonun belirli bir orandaki en zayıf bireyleri popülasyondan çıkartılmakta ve yerlerine arama uzayı üzerinde rastgele oluşturulan yeni bireyler yerleştirilmektedir. Yapılan bu işleme “Injection” adı verilmektedir. Temel amaç, popülasyon içerisindeki genetik çeşitliliği koruyarak ve algoritmanın genel en iyi çözüme ulaşabilme şansının arttırmaktır.

Injection yaklaşımı için her jenerasyonda popülasyonun %20’si kadar yeni birey oluşturulmuş ve popülasyondaki en kötü bireyler ile yer değiştirmişlerdir. Bu yaklaşım algoritmaya dahil edilirken, SGA’nın çalışma adımları üzerinde herhangi bir değişiklik yapılmamış, SGA’nın içerisine yeni bir adım olarak eklenmiştir. Injection yaklaşımının kullanımı Şekil 1’de gösterilmiştir.



Şekil 1: SGA yapısı ve injection yaklaşımının eklenmesi

4. Tartışma

Çalışma sonucunda Injection yaklaşımını SGA ile karşılaştırabilmek ve başarısını gösterebilmek için literatürde yaygın olarak adı geçen sekiz farklı test fonksiyonu kullanılmıştır. Fonksiyonlardan elde edilen test sonuçlarının değerlendirilmesi için ortalama uygunluk değeri ile ortalama uygunluk değerinden faydalanılmıştır. Kullanılan test fonksiyonları, arama uzaylarının büyüklükleri ve minimum noktalar Tablo 1 de verilmiştir.

Tablo 1: Standart Test Fonksiyonları

Fonksiyon	Adı	Arama Uzayı	Minimum Noktası
f_1	Rosenbrock	[-2.048, 2.048]	0
f_2	Rastrigin	[-5.12, 5.12]	0
f_3	Schwefel	[-500, 500]	-418.98n
f_4	Easom	[-100, 100]	-1
f_5	Fifth function of De Jong	[-40, 40]	0.998
f_6	Shubert	[-5.12, 5.12]	-211
f_7	Shekel	[-10, 30]	-3.43
f_8	Michalewicz	[0, π]	-9.66

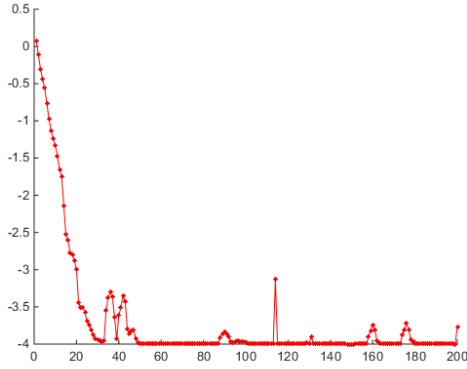
Çalışmanın değerlendirilebilmesi için genetik algoritma içerisinde bulunan birkaç parametrenin önceden ayarlanması gerekmektedir. Popülasyon büyüklüğü, jenerasyon sayısı, mutasyon ve çaprazlama oranları bunlardan birkaçıdır. Bu çalışmada, literatürde standart olarak kabul edilen ve yaygın olarak kullanılan değerler tercih edilmiştir (Tablo 2) [11]. Injection yaklaşımının doğrudan etkisini gösterebilmek için hem SGA hem de Injection yaklaşımı için aynı değerler kullanılmıştır.

Tablo 2: SGA ve Injection yaklaşımı için kullanılan parametreler

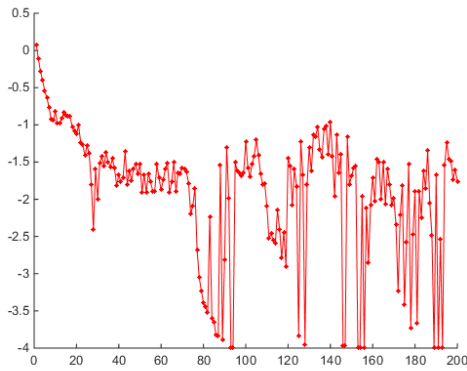
Parametreler	Değerler
Popülasyon büyüklüğü	30
Çaprazlama oranı	75(%)
Mutasyon oranı	1(%)
Turnuva büyüklüğü	3
Jenerasyon sayısı	200
Elit birey oranı	10(%)

Injection yaklaşımının genetik çeşitliliğe nasıl etki ettiğini şekil üzerinde gözlemleyebilmek için Rosenbrock fonksiyonu örnek olarak ele alınmıştır. Genetik çeşitliliği gösterebilmek için standart sapma(SS) değerinden faydalanılmış, değerlerin rassallığın etkisinden arındırılması için fonksiyon her iki metot için 10'ar kez çalıştırılıp her jenerasyondaki ortalama SS değerleri logaritmik ölçekte Şekil 2 ve Şekil 3'te gösterilmiştir. Şekiller incelendiğinde SGA'nın genetik çeşitliliğinin zamanla azalıp sonrasında hep aynı düzeyde kaldığı, Injection yaklaşımı kullanıldığında ise genetik çeşitliliğin zaman zaman azalabildiği ancak çeşitliliğin sürekli olarak yeniden sağlanabildiği görülmektedir.

SGA ile Injection yaklaşımının performanslarının karşılaştırılmasında ise, 200 jenerasyon sonucunda ulaşılan en iyi çözümler ele alınmıştır. Algoritmaların amacına uygun olarak, çözümler arasında daha küçük değere ulaşan, yani minimum değere daha çok yaklaşan çözüm daha başarılı olarak kabul edilmektedir. Sonuçlarının tutarlı olması için her test fonksiyon 10 farklı rastgele başlangıç noktası ile çalıştırılmıştır.



Şekil 2: SGA - Jenerasyonlara göre genetik çeşitliliğin değişimi.



Şekil 3: Injection yaklaşımı - Jenerasyonlara göre genetik çeşitliliğin değişimi.

Tablo 2: SGA ve Injection yaklaşımı için kullanılan parametreler

Fonksiyon	Ortalama Değer		Ortanca Değer	
	SGA	Injection	SGA	Injection
f_1	0,2096	0,0118	0,0663	0,0052
f_2	0,6442	0,0412	0,7242	0,0000
f_3	-701,6306	-837,7942	-719,5258	-837,9655
f_4	-0,4998	-0,8669	-0,4995	-0,9998
f_5	6,2970	1,2958	6,9014	0,9980
f_6	-191,1856	-210,1285	-210,1951	-210,4752
f_7	-2,6859	-3,4263	-2,5029	-3,4332
f_8	-1,7929	-1,8012	-1,8013	-1,8013

SGA ve Injection yaklaşımı için sekiz farklı test fonksiyonu ile elde edilen sonuçlar Tablo 3 te verilmiştir. Verilen tablo, ayrıntılı olarak incelendiğinde, sekiz farklı test fonksiyonunda da Injection yaklaşımının hem ortalama değer hem de orta değer ölçümleri için SGA'dan daha iyi sonuçlar ürettiği gözlemlenmektedir. Bu sonuçlara bakarak, Injection yaklaşımının ek bir adım olarak SGA da kullanılmasının algoritmanın performansını iyileştirdiği söylenebilmektedir.

5. Sonuçlar

Bu çalışmada Injection yaklaşımı ile SGA'da genetik çeşitliliği koruyarak yerel çözümlere erken yakınsama sorununu ortadan kaldırmak ve genetik algoritmanın performansını iyileştirmek amaçlanmıştır. Injection yaklaşımının başarısını değerlendirmek için, literatür de yaygın olarak kullanılan sekiz farklı test fonksiyonundan faydalanılmıştır. Rassallığı ortadan kaldırmak için her test fonksiyonu ile deneyler onar kez tekrarlanmış ve başarı ölçütü olarak ortalama ve ortanca değer ölçü birimleri dikkate alınmıştır. Sonuçlar incelendiğinde Injection yaklaşımı ile SGA'nın performansının arttığı net bir şekilde görülmektedir.

6. Kaynaklar

- [1] Holland, J.H., "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence", Ann Arbor: University Michigan press, 183, 1975.
- [2] Khaji, E. ve Mohammadi, A., S., "A Heuristic Method to Generate Better Initial Population for Evolutionary Methods ", Neural and Evolutionary Computing.
- [3] Ginley, B., M., Maher, J., O'Riordan, C. ve Morgan, F., "Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection", IEEE Transactions on Evolutionary Computation - TEC, 5, 692-714, 2011.
- [4] Friedrich, T., Oliveto, P., S., Sudholt, D. ve Witt C., "Analysis of Diversity-Preserving Mechanisms for Global Exploration", Evolutionary Computation - EC, 4, 455-476, 2009.
- [5] DeJong, K. A., "An analysis of the behavior of a class of genetic adaptative systems", Ph.D. dissertation, Univ. of Michigan, Ann Arbor, 1975.
- [6] SHIMODAIRA, H., "A Diversity Control Oriented Genetic Algorithm", Second International Conference on Genetic Algorithms in Engineering Systems, 444-449, 1997.
- [7] Mauldin, M., "Maintaining genetic diversity in genetic search", National Conference on Artificial Intelligence, 247-250, 1984.
- [8] Winston, P. H., "Artificial Intelligence", Addison-Wesley, 1992.
- [9] P., Snijders, "Incorporating frequency dependent selection and sexual selection in genetic algorithms", Master Thesis 33 pages, 2005.
- [10] Beasley, D., Bull, D. R., Martin, R. R., "An overview of Genetic algorithms: Part I, Fundamentals ", University Computing, 15(2):58-69, 1993.
- [11] Shopova, E.G., Vaklieva- Bancheva, N.G., "BASIC-A Genetic algorithm for engineering problems solution", Computers and Chemical Engineering, 30(8):1293-1309, 2006.