

**GENETİK ALGORİTMA YÖNTEMİYLE TERMOELEKTRİK MODÜLLERİN
OPTİMİZASYONU**

MİTRA SALİMİ ALTAN

**DOKTORA TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**OCAK 2010
ANKARA**

Mitra SALİMİ ALTAN tarafından hazırlanan “Genetik algoritma yöntemiyle termoelektrik modüllerin optimizasyonu” adlı bu tezin doktora tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Raşit AHISKA
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği ile Elektronik- Bilgisayar Eğitimi Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Nihal Fatma GÜLER

Üye : Prof. Dr. Raşit AHISKA

Üye : Prof. Dr. Tofik MAMEDOV

Üye : Doç. Dr. M.Ali AKÇAYOL

Üye : Yr.Doç.Dr. İbrahim SEFA

Tarih : 08 / 11 / 2010

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Mitra SALİMİ ALTAN

**GENETİK ALGORİTMA YÖNTEMİYLE TERMoeLEKTRİK MODÜLLERİN
OPTİMİZASYONU
(DOKTORA TEZİ)**

MİTRA SALİMİ ALTAN

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

Ocak 2010

ÖZET

Bu çalışmada, genetik algoritma yöntemi ile optimizasyon işlemi yaparak soğutma kapasitesi, meydana getirdiği sıcaklık farkı, performans katsayısı maksimum ve maliyeti düşük olan termoelektrik modülleri tasarlamak için 6 farklı program geliştirilmiştir. Bu programların yazılımları Java programlama dili ile NetBeans 7.6.1. ortamında yazılmıştır. Başlangıç popülasyonu olarak 300 adet termoelektrik modül seçilmiştir. Termoelektrik modülün sıcak yüzeyin sıcaklığıyla tanımlanan farklı çalışma rejimleri ve sabit ancak farklı hacimli modüller için modülde kullanılan yarıiletkenin miktarına, kollarının kol alanına, kol yüksekliğine, modüldeki boşlukların toplam hacmine ve kollardan geçen akıma göre modülün optimizasyonu gerçekleştirilmiştir. Optimizasyon sonucunda termoelektrik modülün parametrelerinden maksimum soğutma kapasitesi, performans katsayısı ve maliyetin optimum değerleri elde edilmiştir. Hazırlanan programın uygulaması Melcor firmasının üretmiş olduğu 36 adet termoelektrik modülün her biri için geometrik parametreleri dikkate alınarak yapılmış ve maksimum soğutma kapasiteleri bulunmuştur. Elde edilen sonuçlar ile Melcor firmasının termoelektrik modülleri için öngördüğü soğutma kapasiteleri karşılaştırılmıştır. Karşılaştırılma

sonucunda Melcor firmasının üretmiş olduđu modüllerin sođutma kapasiteleri ile optimizasyon işlemleri ile elde edilen sonuçların birbirlerine çok yakın olduđu görülmüştür.

Bilim Kod : 702.3.006
Anahtar Kelime : Termoelektrik modül, genetik algoritma, maksimum sođutma kapasitesi, optimizasyon
Sayfa Adedi : 112
Tez Yöneticisi : Prof. Dr. Raşit AHISKA

**OPTIMIZATION OF THERMOELECTRIC MODULES WITH THE METHOD
OF GENETIC ALGORITHMS**

(PhD Thesis)

MİTRA SALİMİ ALTAN

**GAZİ UNIVERSITY
INFORMATICS INSTITUTE**

January 2010

ABSTRACT

In this doctoral dissertation, 6 different programs have been developed by employing genetic algorithm method in order to design low-cost thermoelectric modules which maximize cooling capacity, heat difference and the coefficient of performance. The software of these programs has been developed by Java programming language in NetBeans 7.6.1. environment. The initial population selected in this study is 300 thermoelectric modules. Optimization of the module has been carried out depending on different operations regimes of thermoelectric module defined by the heat of the hot surface, the quantity of the semiconductor modules for constant but different volumes modules, the total volume of the space, the arm area, the arm height and the current passing through the arms. As a result of the process of optimization, the optimal values of TE module the coefficient of performance, maximum cooling capacity, and minimum cost have been obtained. The application of the program was performed taking into the geometric parameters for each of the 36 thermoelectric modules produced by Melcor Company and maximum cooling capacities were obtained. Results obtained have been compared with the cooling capacity projected by Melcor Company. The results of the

comparison of the cooling capacity of modules produced by Melcor Company and results obtained by optimization have been found out to be very close.

Science Code : 702.3.006
Key Words : Thermoelectric modules, genetic algorithms, maximizing the cooling capacity, optimization
Page Number : 112
Adviser : Prof. Dr. Raşit AHISKA

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Prof. Dr. Raşit AHISKA'ya tez izleme komitesinde bana sabır gösteren değerli öğretim üyeleri Prof. Dr. Nihal Fatma GÜLER ve Prof. Dr. Tofik MAMEDOV'a ve bütün çalışma sürecinde beni yalnız bırakmayan aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER.....	ix
ÇİZELGELERİN LİSTESİ.....	xiii
ŞEKİLLERİN LİSTESİ.....	xv
SİMGELER VE KISALTMALAR.....	xviii
1. GİRİŞ.....	1
2. TERMOELEKTRİK SİSTEMLER.....	5
2.1. Elektrik, Termoelektrik Olaylar ve Termoelektrik Elemanlar.....	5
2.1.1. Joule olayı.....	5
2.1.2. Termoelektrik olaylar.....	5
2.2. Termoelektrik Modül.....	8
2.2.1. Termoelement.....	8
2.2.2. Termoelektrik modülün yapısı.....	11
2.2.3. Termoelektrik modülün kullanım avantajları.....	13
2.2.4. Termoelektrik modülün uygulama alanları.....	13
2.3. Termoelektrik Yarıiletken ve Modül Parametrelerinin Elde Edilme Yöntemleri.....	13
2.3.1. Termoelektrik yarıiletkenlerin parametreleri.....	13
2.3.2. Termoelektrik modülün performansı.....	14
2.3.3. Klasik yöntem.....	17

	Sayfa
2.3.4. İdealize model.....	18
2.4. Yeni Yöntem.....	19
3. YAPAY ZEKÂ VE GENETİK ALGORİTMA	29
3.1. Optimizasyon Problemlerinin Çözüm Teknikleri ve Genetik Algoritma.....	30
3.2. Genetik Algoritmalar.....	33
3.3. Genetik Algoritmanın Temel Yapısı.....	35
3.3.1. Kromozomların temsili (kodlama).....	36
3.3.2. Başlangıç popülasyonunun oluşturulması.....	37
3.3.3. Amaç fonksiyon veya uyum değeri.....	37
3.3.4. Seçim.....	38
3.3.5. Genetik operatörler.....	41
3.3.5.1. Çaprazlama.....	41
3.3.5.2. Mutasyon.....	45
3.4. Genetik Algoritmaların İşleyişi (Şema Teorisi).....	46
3.5. Genetik Algoritmaların Direk Arama Yöntemlerinden Farkları.....	51
3.6. Genetik Algoritma ve Uygulama Alanları.....	52
4. TERMOELEKTRİK MODELLERİN PARAMETRELERİNİN GENETİK ALGORİTMA İLE OPTİMİZASYONU VE MAKSİMUM SOĞUTMA KAPASİTESİNİN HESAPLANMASI.....	54
4.1. Problemin Genel Tanımı.....	54
4.2. Matematiksel Modelleme.....	55
4.3. Genetik Algoritmanın Uygulanması.....	56
4.3.1. kromozomların oluşturulması.....	56

	Sayfa
4.3.2. Uygunluk fonksiyonunun hesaplanması.....	57
4.3.3. Seçme operatörü.....	58
4.3.4. Çaprazlama.....	59
4.3.5. Mutasyon.....	60
4.3.6. Rulet tekerliği.....	61
4.3.7. Durdurma kriteri.....	61
5 GELİŞTİRİLEN TERMOELEKTRİK MODÜLÜN GENETİK ALGORİTMAYLA OPTİMİZASYON (TEGAO) PROGRAMLARIN YAPISI.....	62
5.1. TEGAO 1 Programının Menüsü ve Çalışması.....	64
5.1.1. TEGAO1 ile maksimum soğutma kapasitesinin hesaplanması.....	64
5.2. TEGAO 2 Programının Menüsü ve Çalışması.....	75
5.2.1. TEGAO 2’de farklı T_H ’ler için maksimum soğutma kapasitesinin hesaplanması.....	75
5.3. TEGAO 3 Programının Menüsü ve Çalışması	77
5.3.1. TEGAO 3 programında farklı T_H ’ler için maksimum soğutma kapasitesi ve maliyetin hesaplanması.....	77
5.4. TEGAO 4 Programının Menüsü ve Çalışması	80
5.4.1. TEGAO 4 farklı ΔT ’ler ve farklı T_H ’ler için maksimum soğutma kapasitesi ve COP hesaplanması.....	80
5.5. TEGAO 5 Programının Menüsü ve Çalışması	82
5.5.1. TEGAO 5 farklı ΔT ’ler için ve farklı boşluklar için maksimum soğutma kapasitesi ve minimum boşlukların hesaplanması ...	82
5.6. TEGAO 6 Programının Menüsü ve Çalışması.....	85

Sayfa

5.6.1. TEGAO 6 farklı ΔT 'ler için ve farklı boşluklar için maksimum soğutma kapasitesi, COP ve minimum maliyetin hesaplanması.....	85
5.7. TEGAO İLE ELDE EDİLEN SONUÇLARIN MELCOR SONUÇLARI İLE KARŞILAŞTIRMASI.....	87
6. SONUÇ VE ÖNERİLER.....	91
KAYNAKLAR.....	94
EKLER.....	99
Ek1 TEGAO programın örnek kodları.....	100
ÖZGEÇMİŞ.....	111

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. TE yarıiletken ve modüllere yönelik parametreler.....	16
Çizelge 3.1. Optimizasyon tekniklerine göre örnekler.....	31
Çizelge 4.1. TEGAO 5'in optimizasyon amaçlı seçtiği 300 adet TE modülün parametrelerinden oluşan örnek listesi.....	57
Çizelge 5.1. TEGAO 1 programında seçilen örnek uygulama kriterleri.....	65
Çizelge 5.2. Rastgele seçilen 300 termoelektrik modülden ilk 30 modülün parametre değerleri ve soğutma kapasiteleri	67
Çizelge 5.3. 50 defa genetik operatör işlemlerinden sonra birkaç termoelektrik modülün parametre değerlerini ve soğutma kapasiteleri	68
Çizelge 5.4. TEGAO 2 programında seçilen örnek uygulama kriterleri.....	76
Çizelge 5.5. TEGAO 2 programında $\Delta T=0$ ve farklı T_H 'ler için elde edilen optimal boyutlar, Q_{CG} ve COP.....	76
Çizelge 5.6. TEGAO 3 programında seçilen örnek uygulama kriterleri.....	78
Çizelge 5.7. TEGAO 3 ile verilen kriterlere göre elde edilen parametre değerleri.....	78
Çizelge 5.8. TEGAO 4 ile verilen parametre değerleri.....	80
Çizelge 5.9 .TEGAO 5 verilen kriterlere göre elde edilen optimal parametre değerleri.....	80
Çizelge 5.10. TEGAO 5 ile verilen parametre değerleri	83
Çizelge 5.11 .TEGAO 5 verilen kriterlere göre elde edilen optimal parametre değerleri.....	83
Çizelge 5.12. TEGAO 6 programında verilen parametre değerleri.....	85
Çizelge 5.13. TEGAO 6 programında verilen kriterlere göre elde edilen optimal parametre değerleri.....	85

Çizelge	Sayfa
Çizelge 5.14. Melcor ile TEGAO programının karşılaştırılma sonuçları.....	88
Çizelge 5.15. Melcor ve TEGAO programı ile elde edilen modüllerin soğutma kapasitelerinin geometrik faktöre göre dağılımı	89

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Seebeck olayını açıklayan devre.....	6
Şekil 2.2. Peltier olayını açıklayan devre.....	7
Şekil 2.3. Thomson olayını açıklayan devre.....	8
Şekil 2.4. Termoelementin yapısı.....	9
Şekil 2.5. Termoelement üzerinde oluşan ısı yükler.....	11
Şekil 2.6. Termoelektrik modülün görüntüsü ve yapısı.....	12
Şekil 2.7. Termoelektrik modülün jeneratör olarak çalışması.....	12
Şekil 2.8. TE modülün giriş ve çıkış parametreleri.....	15
Şekil 3.1. Genetik algoritmanın akış şeması.....	36
Şekil 3.2. Tek nokta çaprazlama.....	42
Şekil 3.3. İki nokta çaprazlama.....	43
Şekil 3.4. Düzgün çaprazlama	44
Şekil 3.5. Mutasyon	45
Şekil 3.6. Genetik algoritmaların genel yapısı.....	46
Şekil 4.1. İki TE modülün parametreleri arasında tek nokta çaprazlama.....	60
Şekil 5.1. TEGAO 1, 2 ve 3 program menüsünün görüntüsü.....	62
Şekil 5.2. TEGAO 4, 5 ve 6 program menüsünün görüntüsü.....	63
Şekil 5.3. Parametre değerleri girilmiş TEGAO 1 programın menüsü	65
Şekil 5.4. TEGAO1 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, maksimum soğutma kapasitesi, maksimum akım ve optimal kol alanın görüntüsü.....	69
Şekil 5.5. TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü.....	70

Şekil	Sayfa
Şekil 5.6. TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü.....	70
Şekil 5.7. TEGAO 1 program ile Q_C 'nin 50 iterasyon (turuna) göre yaklaşımı.....	71
Şekil 5. 8. TEGAO 1 program ile akımın 50 iterasyona (turuna) göre Yaklaşımı.....	72
Şekil 5. 9. TEGAO 1 program ile alanın 50 iterasyona (turuna) göre yaklaşımı.....	72
Şekil 5. 10. TEGAO 1 program ile kol yüksekliğinin 50 iterasyona (turuna) göre yaklaşımı.....	73
Şekil 5. 11. TEGAO 1 program ile kol sayısının 50 iterasyona (turuna) göre yaklaşımı.....	73
Şekil 5.12. TEGAO 1 program ile Q_C 'nin akıma göre değişimi.....	74
Şekil 5.13. TEGAO 1 program ile Q_C 'nin kol alanına göre değişimi.....	74
Şekil 5.14. TEGAO 1 program ile Q_C 'nin kol yüksekliğine göre değişimi.....	74
Şekil 5.15. Farklı T_H 'ler için Q_C 'nin değerleri.....	77
Şekil 5.16. TEGAO 3 ile Q_C , akım ve alanın 50 iterasyona göre yaklaşımı.....	78
Şekil 5.17. TEGAO 3 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre ekran görüntüsü.....	79
Şekil 5.18. TEGAO 3 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre ekran görüntüsü.....	79
Şekil 5.19. TEGAO 4 ile Q_C , akım ve alanın 50 iterasyona göre yaklaşımı.....	81
Şekil 5.20. TEGAO 4 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü.....	81

Şekil	Sayfa
Şekil 5.21. TEGAO 4 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü.....	82
Şekil 5.22. TEGAO 5 ile Q_C , akım ve alanın 50 iterasyon göre yaklaşımı.....	83
Şekil 5.23. TEGAO 5 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü.....	84
Şekil 5.24. TEGAO 5 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü.....	84
Şekil 5.25. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, maksimum soğutma kapasitesi, maksimum akım ve optimal kol alanın görüntüsü.....	86
Şekil 5.26. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü.....	86
Şekil 5.27. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü.....	87
Şekil 5.28. Q_{CM} , Q_{CG} ve akımın geometrik faktöre göre değişimi.....	90

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
A	Amper
°C	Santigrad Derece
F	Fahrenhayd
I	Akım
J	Joule
K	Kelvin
kg	Kilogram
R	Direnç
Q_j	Joule ısı yükü
Q_P	Peltier Isısı
Q_T	Thomson Isısı
V	Volt
ΔT	Sıcaklık Farkı
Q_C	Soğuk Yüzeyin Soğutma Kapasitesi
Q_H	Sıcak Yüzeyden Emilen Isı Miktarı
N	Kolların Sayısı
k	Isı İletkenlik
H	Kol Yüksekliği
T_C	Soğuk Yüzeyin sıcaklığı
T_H	Sıcak yüzeyin Sıcaklığı
G	Geometri Factor
α	Seebeck Katsayısı
d	Termoelektrik Materyalin Yoğunluğu

Simgeler	Açıklama
ρ_r	Termoelektrik Materyalin Elektriksel Özdirenç
r_c	Elektriksel Kontak Direnç
P	TE Modülün Tükettiği Güç
T_{mid}	Sıcaklık Ortalaması
δ	Kollar Arasındaki Boşluk

Kısaltmalar	Açıklama
A	Alan
COP	Performans Katsayı
DC	Doğru Akım
EŞ	Eşitlik
GA	Genetik Algoritma
N	Negatif
P	Pozitif
TE	Termoelektrik
TEGAO	Termoelektrik Modülün Genetik Algoritmayla Optimizasyonu
YZ	Yapay Zekâ

1. GİRİŞ

Bir termoelektrik cihazın tasarımında, kullanılacak termoelektrik modülün (TE) doğru seçilmesi çok önemlidir. Modüllerin seçilmesinde en önemli faktörler ise TE modüllerin maksimum soğutma kapasitesi, sıcaklık farklılığı, performans katsayısı ve maliyeti olarak sıralanabilir. Dolayısıyla gerekli TE'lerin üretilmesi için modüller belirli bir soğutma kapasitesi, meydana getirdikleri sıcaklık farklılığı, performans katsayısı ve maliyetine göre tasarlanmalıdır. Termoelektrik modüllerin kollarının boyutları (kol yüksekliği, kol alanı, kol sayısı), kollardan geçen akım optimize edilerek üretime geçmeden daha yüksek bir soğutma kapasiteli, sıcaklık farklılığı, performans katsayısı ve daha düşük maliyetli modüller bir TE'n tasarımı sağlanır [1-11].

Son zamanlarda optimizasyon tekniği olarak genetik algoritmalar kullanılmakta ve çok karmaşık problemlerin çözümünde yaygın olarak uygulanmaktadır. Genetik algoritmalar, doğal seçme ve doğal genetiğin evrim prensiplerine dayalı optimizasyon tekniği ve bilgisayar destekli arama tekniği olarak kullanılmaktadır. İşlemler, olası çözümlerin bir popülasyonundan başlar ve sonra en uygunları seçilir. Optimizasyonu gerçekleştirmek için, temel genetik işlemlerden olan seçme, çaprazlama ve mutasyon uygulanır [12-22].

Bu çalışmada, termoelektrik modüllerin maksimum soğutma kapasitesi (Q_{Cmax}), sıcaklık farklılığı (ΔT), performans katsayısı (COP), güç (P) ve düşük maliyet elde etmek için genetik algoritma yöntemini kullanarak 6 farklı program geliştirilmiştir. Başlangıç popülasyon olarak 300 adet termoelektrik modül seçilmiştir. Farklı sıcak yüzeyin sıcaklığı (T_H) için modülün yapımında kullanılan yarıiletkenlerin miktarına, kolların ve boşlukların toplam hacmine (V), kol alanına (A), kol yüksekliğine (H), kol sayısına (N) ve kollardan geçen akıma (I) göre termoelektrik modülün optimizasyon yapılmıştır. Genetik algoritma yöntemi ile optimizasyon işlemi yapılarak TE modülün tüm parametrelerin optimal değerleri ve maliyeti elde edilmiştir. Ayrıca Melcor

firmasının üretmiş olduğu 36 adet termoelektrik modülün her biri için geometrik parametrelerini dikkate alarak termoelektrik modüller için maksimum soğutma kapasitesi elde edilmiştir [23].

Bu çalışmada termoelektrik modülün genetik algoritmayla optimizasyon (TEGAO) programının 6 farklı versiyonda yazılmıştır. Bu programların yazılımları Java programlama dili ile NetBeans 7.6.1. ortamında geliştirilmiştir.

TEGAO 1 programı ile modülün hacmi sınırlı ve sıcaklık farkı sıfır, sıcak yüzeyin sıcaklığı 300K kabul edilerek soğutma kapasitesi ve performans katsayısının maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır.

TEGAO 2 programı ile modülün hacmi sınırlı ve sıcaklık farkı sıfır ve sıfırdan farklı kabul edilerek farklı sıcaklıklar için soğutma kapasitesi (Q_C) ve COP'un maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır.

TEGAO 3 programı ile modülün hacmi sınırlı ve ΔT sıfır ve sıfırdan farklı kabul edilerek değişik T_H 'ler için soğutma kapasitesi ve COP'un maksimum değerlerinin ve maliyetin minimum elde edilmesi için kol alanı, kol yüksekliği ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır.

TEGAO 4 programı ile modülün hacmi sınırlı ve sıcak yüzeyin sıcaklığı, sıcaklık farkı ve kollar arasındaki boşluk alanı farklı kabul edilerek soğutma kapasitesi, güç ve COP'un maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk, sıcak yüzeyin sıcaklığı ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır. Burada sadece T_H 'nin etkisi değil, kollar arasındaki boşluğunda, Q_C , P ve COP (Coefficient of Performance) üzerindeki etkisi araştırılmıştır.

TEGAO 5 programı ile modülün hacmi sınırlı kabul edilerek soğutma kapasitesi, performans katsayısı ve ΔT 'nin sıfırla maksimum değerleri arasındaki sayılarının elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır. Bunun sonucunda modülün Q_{Cmax} ve ΔT_{max} çalışma rejimlerinin tespit edilmesi sağlanmıştır.

TEGAO 6 programı ile modülün hacmi sınırlı, T_H ve ΔT farklı kabul edilerek soğutma kapasitesi, COP maksimum ve maliyetin minimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve kollardan geçen akıma göre optimizasyon işlemi yapılmıştır. Burada sadece T_H 'nin etkisi değil, kollar arasındaki boşluğun, Q_C , P , COP ve maliyet üzerindeki etkisi araştırılmıştır.

Tezin ikinci bölümünde termoelektrik sistemler ele alınmıştır. Joule ve termoelektrik olaylar ve termoelektrik elemanlar tanımlanmıştır. Termoelektrik yarıiletkenler ve bunlardan yapılan termoelementler ve modüller genel olarak incelenmiştir ve termoelektrik modüllerin avantajları açıklanmıştır. Yine bu bölümde termoelektrik modülün parametreleri tanımlanmış olup bu parametrelerin elde edilmesi için klasik ve yeni yöntemler ayrıntılı bir biçimde incelenmiştir.

Üçüncü bölümde yapay zekâ teknikleri ile ilgili genel bilgiler sunulmuştur. Burada ağırlık olarak tezde kullanılan yapay zeka optimizasyon tekniklerinden, genetik algoritma konusu ayrıntılı olarak incelenmiştir. Genetik algoritmayı hakkında genel bilgiler verildikten sonra genetik algoritmanın işleyiş sürecine ve operatörlerine yer verilmiştir.

Dördüncü bölümde genetik algoritmanın yönteminin de optimizasyon için ilgili temel bilgiler verilmiştir. Burada matematiksel modelleme ve genetik algoritma uygulanması ayrıntı olarak ele alınmıştır.

Beşinci bölümde geliştirilen termoelektrik modülün genetik algoritmayla optimizasyon TEGAO program ele alınmıştır. Burada TEGAO yapısı incelenmiştir ve 6 farklı program ayrı ayrı ele alınmıştır ve örnekler gösterilmiştir. TMGAO ile elde edilen sonuçlar Melcor firmasının üretmiş olduğu 36 adet CP serisiyle karşılaştırılmıştır.

Altıncı bölümde sonuç ve öneriler verilmiştir.

2. TERMoeLEKTRİK SİSTEMLER

2.1. Elektrik, Termoelektrik Olaylar ve Termoelektrik Elemanlar

Herhangi bir maddeyi karakterize eden, uygulama açısından da önemli olan, maddenin elektriksel özellikleridir. Elektriksel özellikleri karakterize eden Joule ve termoelektrik olaylarıdır.

2.1.1. Joule olayı

James Prescott Joule'un elektrik enerjisinin ısı enerjisine dönüşümünü ifade eden ve kendi adı ile isimlendirilen kanununa göre, akım taşıyan bir iletken, direnci ve içinden geçen akımın karesiyle orantılı olarak ısını artırır. Bir elektrik devresinde birim zamanda ortaya çıkan Joule ısı miktarı (Watt olarak) Eş. 2.1 ile ifade edilir.

$$Q_j = I^2 \cdot R \quad (2.1)$$

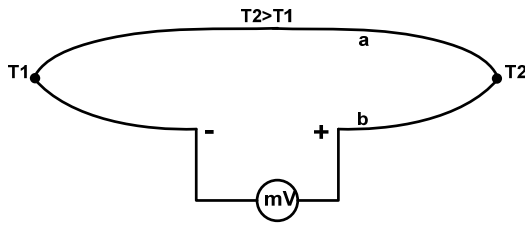
Burada Q_j , devreden geçen I akımının etkisi ile ortaya çıkan toplam Joule ısı yükünü, I devreden geçen akım miktarını, R ise devrenin elektrik akımına gösterdiği toplam direnç miktarını ifade etmektedir.

2.1.2. Termoelektrik olaylar

Isı enerjisi ile elektrik enerjisinin birbirlerine doğrudan dönüşümünü termoelektrik olayını tanımlar. Bu tanım, İngiliz fizikçi James P. Joule'ün tanımladığı geri dönüşümü olmayan elektrik-ısı dönüşümü ile Seebeck, Peltier ve Thomson tarafından keşfedilen ve kendi adlarıyla anılan olaylara termoelektrik olaylar denir.

Seebeck Etkisi: Alman fizikçi Thomas J. Seebeck 1820'de, farklı metallere yapılmış iki tel parçasının uçlarını birleştirerek bir kapalı devre meydana

getirildiğinde ve metallerin birleşim noktaları arasında bir sıcaklık farkı meydana getirildiğinde, sıcaklık farkıyla doğru orantılı bir gerilim meydana geldiğini bulmuştur. Böyle bir devre “termokupl” olarak adlandırılır. Şekil 2.1’de gösterildiği üzere a ve b gibi farklı maddelerden oluşan ve temas uçları arasında sıcaklık farkı ($\Delta T = T_2 - T_1$) meydana getirildiğinde bu uçlar arasında potansiyel farkı (ΔV) meydana gelir.



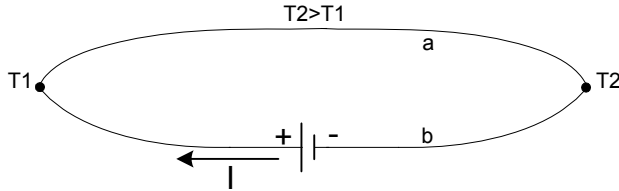
Şekil 2.1. Seebeck olayını açıklayan devre

$$\Delta V = \alpha \cdot \Delta T \quad (2.2)$$

Burada α Seebeck katsayısını veya termoemk'yı ifade eder. Birimi $V / ^\circ C$ 'dır. α 'nın değeri devreyi oluşturan maddelerin özelliklerine bağlıdır. Örneğin bakır konstantan'dan yapılan bir termokupl için $\alpha = 40 \mu V / ^\circ C$ 'dır. Yani her $1^\circ C$ sıcaklık farkı için $40 \mu V$ 'luk bir emk üretir. Yarıiletkenlerde bu değerler çok daha yüksektir. α 'nın değeri $100 \mu V / ^\circ C$ 'den büyük olan yarıiletkenlere “termoelektrik yarıiletkenler” denir. Yarıiletkenlerin tipine göre α 'nın işareti değişmektedir. N tipi yarıiletkenler için α negatif, p tipi yarıiletkenler için ise pozitifdir. Metallerde Seebeck olayı, termokupl veya ısı algılayıcısı gibi, termoelektrik yarıiletkenlerde ise jeneratör gibi kullanılır.

Peltier Etkisi: 1834’de Fransız fizikçi Jean C. A. Peltier (1785–1845), Seebeck etkisinin tersi bir durum keşfetti. Eğer termokupl içerisinde bir akım geçerse, Joule ısı ile birlikte bir eklem noktasından ötekisine ısı transferi sağlayacak şekilde bir ucun sıcaklığı düşerken ötekisinin sıcaklığı yükselir. Joule ısısından farklı olarak ortaya çıkan bu ısıya *Peltier ısısı* denir. Açığa çıkan ısı miktarı, uygulanan akım şiddetiyle orantılıdır ve yönü akım yönünün

değiştirilmesi ile değişir. Peltier olayını açıklayan devre Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Peltier olayını açıklayan devre

Birim zamanda üretilen Peltier ısısı (Watt olarak) Eş. 2.3 ile ifade edilir.

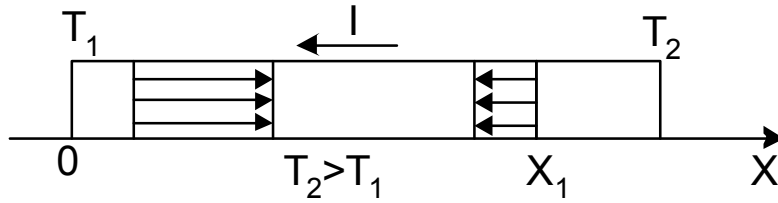
$$Q_p = \pi I \quad (2.3)$$

Burada π , Peltier katsayısını ifade eder.

Thomson etkisi: İskoç bilim adamı William Thomson (daha sonraları Lord Kelvin olarak anılacaktır), 1856'da termoemk (α) ile Peltier katsayısı (π) arasındaki bağıntıyı kaydetmiştir (Eş. 2.4).

$$\alpha = \frac{\pi}{T(^{\circ}K)} \quad (2.4)$$

Buna ek olarak kendi ismini taşıyacak olan yeni bir olayın hipotezini ortaya atmıştır. Bu olay 1867'de Leru tarafından deneysel olarak kanıtlanmıştır. Thomson olayı şöyle açıklanabilir. Şekil 2.3'de görüldüğü gibi, akım taşıyan bir iletkenin herhangi iki noktası arasında bir sıcaklık farkı varsa akım yönüne göre iletkende Joule ısısına ek olarak Thomson ısısı (Q_T) açığa çıkmaktadır.



Şekil 2.3. Thomson olayını açıklayan devre

Birim zamanda ortaya çıkan Thomson ısı, akım şiddeti (I), sıcaklık farkı (ΔT) doğru orantılıdır.

$$Q_T = I \cdot \alpha_T \cdot \Delta T = \alpha_T \cdot (T_2 - T_1) \quad (2.5)$$

İki farklı metalden oluşan bir devre için Thomson termo emk'ları arasındaki ilişki aşağıdaki bağıntıda ifade edilmiştir.

$$\alpha_T = (\alpha_{Ta} - \alpha_{Tb}) \frac{T_2 - T_1}{T_2} \quad (2.6)$$

Buradaki α_T Thomson olayında meydana gelen termoemk'dır. Formüldeki α_{Ta} ve α_{Tb} sayıları farklı maddelerden oluşan devreyi ifade etmek için kullanılmaktadır. Termoelektrik olaylar oluşan devrelerde Thomson olayını yok etmek için α_T değerleri eşit olan malzemeler seçilmelidir.

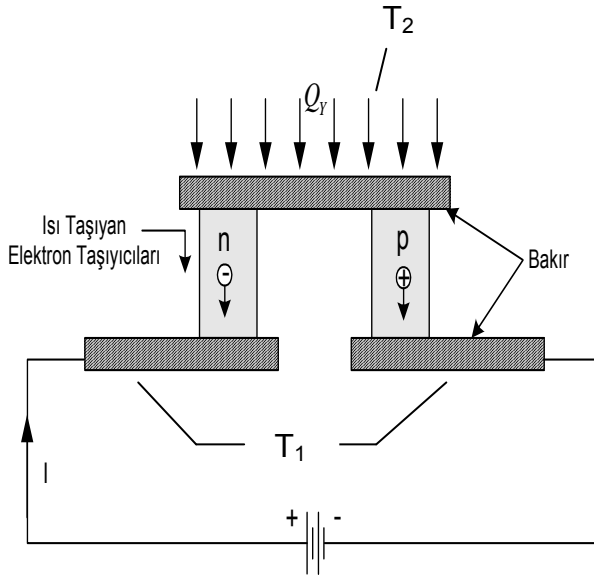
2.2. Termoelektrik Modül

2.2.1. Termoelement

Herhangi bir termoelektrik cihaz veya devrenin temelinde termoelementlerden oluşan termoelektrik modül bulunmaktadır. Bir termoelektrik modül termoelementlerin elektriksel olarak seri ve termal olarak paralel bağlanmasıyla meydana gelir.

Bir termoelement, n ve p tipi yarıiletkenlerin, Şekil 2.4'de gösterildiği gibi, bir iletkenle (Bakır veya alüminyum) elektriksel olarak seri bağlanmasıyla oluşur.

Burada n tipi yarıiletken termoelement'in negatif, p tipi yarıiletkende termoelementin pozitif kolları olarak tanımlanır. P ve N tipi termoelementler de kullanılan yarıiletken malzemeler; Bi_2Te_3 , PbTe, SiGe ve BiSb gibi alaşımlardır. Bizmut Teleryum (Bi_2Te_3), yüksek termoelektrik verimi ve uygun sıcaklık aralığı nedeniyle yaygın bir şekilde tercih edilmektedir. Alt ve üst plakalarda seramik malzeme kullanılır. Böylece ısıl iletkenlik, elektriksel yalıtkanlık ve mekanik mukavemet sağlanır. Bir termoelementin çalışma prensibine bakılırsa, eğer termoelemente Şekil 2.4'deki gibi DC bir gerilim uygulanırsa, yük taşıyıcıları, enerji seviyesi düşük olan üstteki bakırdan yarıiletkene geçerler. Bakır ile yarıiletken arasında mevcut olan enerji duvarını aşmak için elektron ve delikler bakırın örgüsünden enerji sağlarlar. Böylece üstteki bakırın sıcaklığı düşecektir. Alttaki bakıra gelince, yüksek enerji seviyesinden düşük enerji seviyesine geçen yük taşıyıcıları sahip oldukları fazla enerjiyi alttaki iletkene aktararak ısınmasına neden olacaktır. Bu geçiş sırasında soğuyan yüzeyden ısı emilirken soğuk yüzeyin sıcaklığı düşecek, ısının pompalandığı sıcak yüzeyin sıcaklığı ise artacaktır. Devreye uygulanan akımın yönü değiştirilirse soğuyan yüzey ısınır ısınan yüzey soğur.

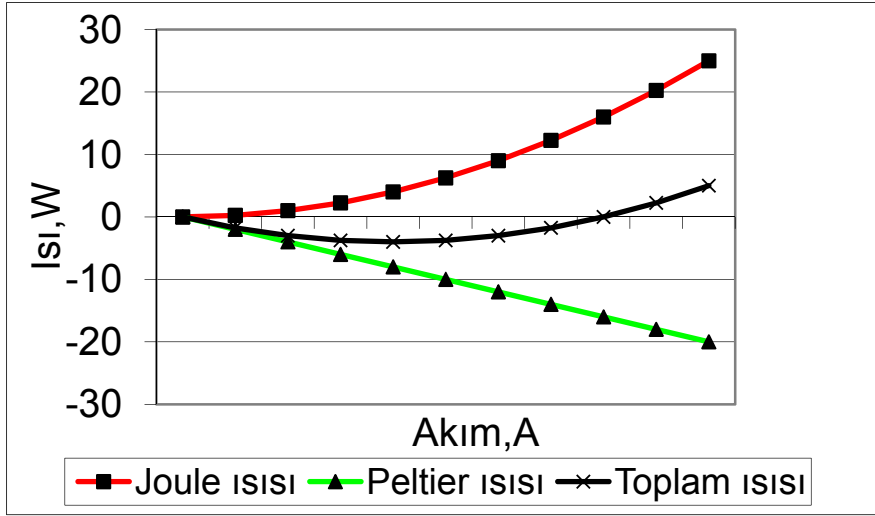


Şekil 2.4. Termoelementin yapısı

Şekil 2.4'de gösterildiği gibi, eğer çalışan bir termoelementin sıcak yüzeyinde açığa çıkan ısıyı herhangi bir ısı transfer sistemiyle çevreye yayarak T_1 sıcaklığı sabit tutulursa devreden geçen I akım şiddetine bağlı olarak soğuk yüzey sıcaklığı belli bir T_2 değerine kadar düşer. Bir termoelement üzerinden geçen akım şiddeti sabit tutulduğunda T_2 sıcaklığının değeri soğuk yüzeye gelen ve emilen Q_y ısı yüküne bağlıdır. Bu ısı yükü çevreden ve sıcak levhadan soğuk levhaya ısı iletiminden dolayı gelen ve termoelement devresinden geçen akım şiddeti etkisi ile açığa çıkan Joule ısılarından oluşur.

Şekil 2.5'de bir termoelementin soğuyan levhasına düşen Joule (Q_J) ve Peltier (Q_P) ısılarını ve bu ısıların ($Q = Q_J + Q_P$) toplamının akım şiddetine (I) göre değişim grafikleri gösterilmiştir. Peltier ve Joule olayları aynı elektrik devresinde gerçekleştiğinden dolayı uygulanan herhangi bir akım şiddeti için bir termoelementin ısısal dengesi Q_P ve Q_J 'nin cebirsel toplamı olan $Q = Q(I)$ fonksiyonu ile karakterize edilmektedir.

Şekil 2.5'deki $Q = Q(I)$ grafiği bir minimumdan geçmektedir. Toplam ısının minimuma ulaştığı akım şiddetine maksimum akım (I_{max}) denir. Bir termoelemente maksimum akım uygulandığında soğuyan levha üzerindeki sıcaklık düşümü maksimum olur. Bir termoelementin veya termoelementlerden oluşan bir modülün maksimum akım şiddetinin değeri termoelementin üretiminde kullanılan termoelektrik yarıiletkenlerin kalitesine, boyutlarına ve yapısal özelliklerine göre değişir. Bir termoelemente uygulanan akım şiddeti, I_{max} 'dan büyük olduğunda Joule ısı Peltier ısısından fazla olur. Bu nedenle soğuma etkisi azalır ve akımı maksimum akımın iki katına ($I=2I_{max}$)'a ulaştığında soğutma gerçekleşmez. Bu durumda Peltier ısı Joule ısını dengeler. Akım değeri daha da arttığında, yani $I > I_{max}$ olduğundan ısısal denge bozulur. Joule ısı Peltier ısısından daha fazla olur ve termoelement soğutma özelliğini kaybeder. Bu durumda termoelement bir rezistans gibi davranır ve ısınır

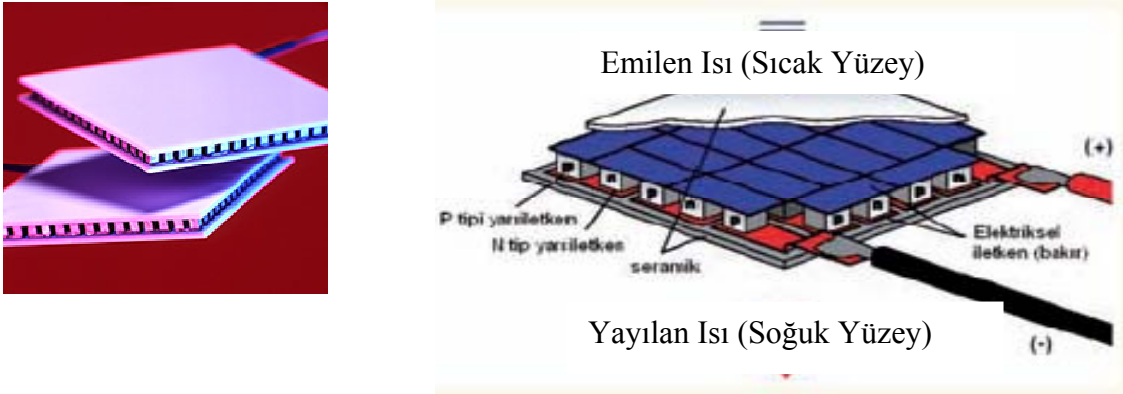


Şekil 2.5. Termoelement üzerinde oluşan ısı yükler

Bir termoelementin çalışmasını en çok etkileyen Joule ısıdır. Peltier ısı akım şiddeti ile doğru orantılı iken Joule ısı akım şiddetinin karesi ile doğru orantılıdır. Bir yaklaşım olarak, termoelementin üzerinde açığa çıkan Joule ısının levhalar arasında eşit paylaştığını kabul edersek soğuyan levhaya düşen ısı miktarı Joule ısının yarısıdır [2,10,24].

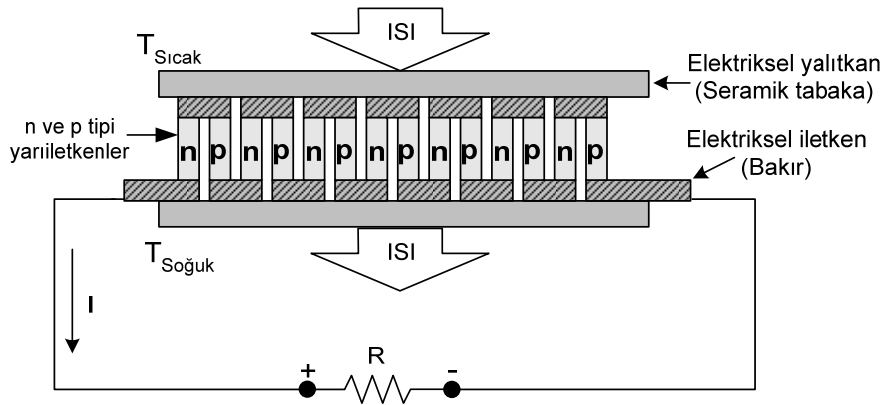
2.2.2. Termoelektrik modülün yapısı

Onlarca termoelement, elektriksel olarak seri termal olarak paralel, bağlanarak çeşitli amaçlar için çeşitli büyüklüklerde termoelektrik modüller elde edilmektedir. Şekil 2.6'de termoelektrik modülün görüntüsü ve yapısını göstermektedir [25-26].



Şekil 2.6. Termoelektrik modülün görüntüsü ve yapısı

Termoelektrik modüller sadece soğutma ısıtma amaçlı değil aynı zamanda enerji üretiminde de kullanılmaktadır. Eğer termoelektrik modülün iki yüzeyi arasında sıcaklık farkı meydana getirilirse o zaman modülde meydana gelen Seebeck olayından dolayı, modül bir DC güç kaynağı gibi davranır. Mevcut olan güneş pillerinden farklı olarak termoelektrik jeneratörler yüksek derecede akım ve orta derecede gerilim üretebilmektedir. Termoelektrik modülün jeneratör olarak çalışmasını sağlayan devre Şekil 2.7’de gösterilmiştir.



Şekil 2.7. Termoelektrik modülün jeneratör olarak çalışması

2.2.3. Termoelektrik modülün kullanım avantajları

Bir termoelektrik soğutma modülü, peltier olayını kullanan bir ısı pompasıdır. Bu etki modül içinden modülün bir tarafından diğer tarafına geçen ısı transferine kılavuzluk eden doğru akımının bir oluşumudur. Sonuç olarak, modülün bir tarafı soğur, diğer tarafı ısınır.

Termoelektrik modüllerinin kullanımı ile aşağıdaki avantajlar elde edilebilir.

- Sesi ve titreşimi yoktur.
- Küçük, hafif ve basit yapıdadır.
- Akım akış yönünün basit değişimi yoluyla soğuyan ve ısınan yüzeyler değişir. Soğuyan yüzey ısınır ve ısınan yüzey soğur.
- Sıcaklık kontrolleri kolaylıkla yapılabilmektedir.

2.2.4. Termoelektrik modülün uygulama alanları

Termoelektrik uygulamalarını askeri/uzay gemisi, tüketici ürünleri, laboratuvar ve bilimsel teçhizatlar gibi üç ana başlık altında toplamak mümkündür [27].

2.3. Termoelektrik Yarıiletken ve Modül Parametrelerinin Elde Edilme Yöntemleri

2.3.1. Termoelektrik yarıiletkenlerin parametreleri

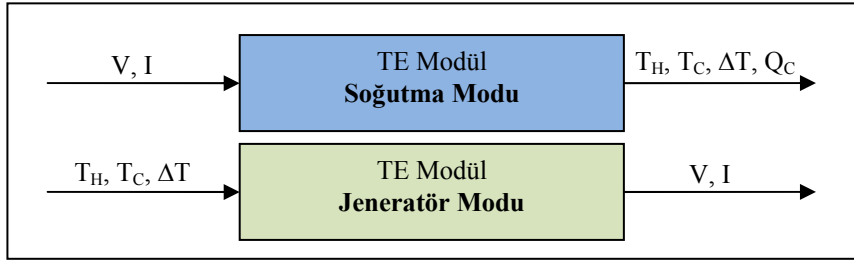
Hem teorik araştırmalar açısından hem de deneysel araştırmalar açısından TE yarıiletkenlerin α , ρ , κ ve z parametrelerinin ölçülmesi büyük önem taşımaktadır. Bu parametreler TE modül ve sistemlerin kalitesini belirlemektedir. TE yarıiletkenlerle üretilen modüllerin ve bu modülleri kullanarak geliştirilen TE sistemlerin verimi Z parametresine bağlıdır. TE

modülün Z parametre değeri ne kadar büyükse, TE yarıiletkenin performansı da o kadar iyi olmaktadır. Bu nedenle Z parametresini belirleyen tüm parametrelerin ölçülmesi gerekmektedir. Günümüz teknolojisinde Bi_2TE_3 ve alaşımlarının Z değeri en yüksek olduğundan TE sistemlerde kullanılan en uygun TE malzeme durumundadır. TE yarıiletkenlerin temel parametreleri sıcaklığa göre değişmektedir. Bu değişimlerin araştırılması yarıiletkenlerin temel araştırmalarını oluşturmaktadır. TE modül ve sistemlerin çeşitli sıcaklık ortamlarında çalışması gerektiğinden bu parametrelerin sıcaklığa bağlı değişimlerinin araştırılması da büyük önem taşımaktadır.

Sıcaklığa bağlı iletkenlik metallerde doğrusal olarak değişirken yarıiletkenlerde ise bu değişim doğrusal değildir. TE yarıiletkene DC akım uygulandığında meydana gelen termoemk geriliminden dolayı iletkenlik değişmektedir. Bu nedenle AC uygulayarak ölçümlerin yapılması daha uygun olmaktadır. Ancak Joule ısısından dolayı AC akımda bile TE yarıiletkenlerin iç direnci değişmektedir. Bu nedenle direnç ölçmede Ohm Kanunu doğrudan kullanılamamaktadır [28-38].

2.3.2. Termoelektrik modülün performansı

TE modüller yapıları gereği hem soğutma hem de jeneratör modunda çalışabilmektedirler. Soğutma modunda TE modüller için giriş parametresi akım (I) ve gerilim (V) iken çıkış parametresi ısınan yüzey (T_H) ve soğuyan yüzey (T_C) arasındaki sıcaklık farkı (ΔT) olmaktadır. Jeneratör modunda ise giriş parametresi $\Delta T = T_H - T_C$ iken çıkış parametresi I ve V'dir (Şekil 2.8). TE modüllerin performansı uygulanan giriş parametresine (V ve I) bağlı olarak yüzeyler arasında elde edilebilecek maksimum sıcaklık farkı (ΔT) veya TE modülün soğuk yüzeyinden emilen maksimum ısı yükü (Q_C - Soğutma Gücü) olarak tanımlanabilir. TE modülün performansı aynı zamanda onun ısı pompalama kapasitesi olarak da ifade edilebilmektedir. Teknolojik olarak ticari TE jeneratörler, TE soğutuculara göre genelde azdır. TE modülün giriş ve çıkış parametreleri Şekil 2.8'de gösterilmektedir.



Şekil 2.8. TE modülün giriş ve çıkış parametreleri

TE modülün yüzeyleri arasında elde edilecek sıcaklık farkının belirlenmesinde uygulanan akım, ısınan yüzey sıcaklığı ve soğuyan yüzeye etki eden ısı yük miktarı büyük rol oynamaktadır. TE modülün ürettiği termoemk ise sıcaklık farkıyla orantılı olarak artmaktadır. TE modül performansını etkileyen en önemli faktör uygulanan akım şiddeti miktarıdır. Ayrıca TE modülün performansının belirlenmesinde T_H değerinin de doğrudan etkisi bulunmaktadır. Çünkü TE modülün T_H değerindeki artış TE modülde 'iç ısı yük' etkisi meydana getirdiğinden buna bağlı olarak T_C değeri de yükselmektedir. T_H değerini, çevre sıcaklığı ve ısı transfer sisteminin performansı belirlemektedir. Bu nedenle TE sistemlerde, ısı transfer sistemleri kullanılarak T_H değeri düşürülmekte ve bu sayede de TE modüllerin soğutma performansları artırılabilir. Bu nedenlerden dolayı TE modüllerin seçiminde T_H , T_C , ΔT ve Q_C parametrelerinin belirlenmesi büyük önem taşımaktadır [1-10]. Sıcaklık farklarında, maksimum COP ya da soğutma verimine sahiptirler. Buna karşılık jeneratörler büyük sıcaklık farklarında maksimum verime sahiptirler. TE jeneratörlerin yüksek işletim sıcaklıkları sonucunda, PbTe ve SiGe alaşımları gibi farklı maddeler ve tipik soğutuculara göre farklı derleme teknolojilerine ihtiyaç duyulmaktadır [29,30].

TE modüllerin performans ve çalışma şartlarını belirleyen çeşitli parametreler mevcuttur. TE yarıiletken ve modüllere yönelik parametreler Çizelge 2.1'de özetlenmiştir. TE modüle uygulanan DC akım parametresine bağlı olarak Q_C , T_C , P , COP , K , R , Z çıkış parametreleri ile V_{max} , I_{max} ve E_{max} olarak adlandırılan deneysel parametreleri mevcuttur.

Çizelge 2.1. TE yarıiletken ve modüllere yönelik parametreler

Simge	Birim	Açıklama
COP		TE modül Performans Katsayısı
E_{max}	V	TE modülün ürettiği termoemk (ΔT_{max} iken)
I_{max}	A	TE modülün çektiği akım (ΔT_{max} iken)
K	$W.K^{-1}$	TE modülün toplam termal iletkenliği
P	W	TE modülün tükettiği güç
Q_C	W	Soğuk yüzeyin soğutma kapasitesi
Q_H	W	Sıcak yüzeyin ısıtma kapasitesi
Q_L	W	Birim zamanda TE modüle gelen toplam ısı yük
Q_{Cmax}	W	TE modülün maksimum soğutma kapasitesi ($I = I_{max}$ ve $\Delta T = 0$ iken)
R	Ohm	TE modül elektriksel direnci
T_C	K	TE modül soğuk yüzey sıcaklığı
T_H	K	TE modül sıcak yüzey sıcaklığı
V_{max}	V	TE modül gerilimi (ΔT_{max} iken)
Z	K^{-1}	TE modülün kalite katsayısı (Figure-of-merit)
ΔT	K	TE modül yüzeylerarası sıcaklık farkı ($\Delta T = T_H - T_C$)
ΔT_{max}	K	TE modülde elde edilebilecek maksimum sıcaklık farkı ($I = I_{max}$ ve $Q_C = 0$ iken)
κ	$W.m^{-1}.K^{-1}$	TE yarıiletkenin termal iletkenliği
α	$V.K^{-1}$	TE yarıiletkenin Seebeck katsayısı
ρ	Ohm.cm	TE yarıiletkenin öz direnci
z	K^{-1}	TE yarıiletken kalite katsayısı

TE modüllerin performans analizlerinin yapılması ve parametrelerinin elde edilmesi amacıyla klasik hale gelmiş çeşitli yöntemler geliştirilmiştir. Ayrıca bu yöntemlerin iyileştirilmesi amacıyla Melcor tarafından geliştirilen yöntemler de mevcuttur [24]. Ancak bu yöntemler yarıiletkenlerin parametrelerini sabit kabul etmekte veya sadece sıcaklık değişimlerini öngörmektedir. Bu nedenle; bu yöntemlerle, çalışır durumdaki TE modülün performans hesaplamaları ve

çıkış parametrelerinin değerlendirilmesi hatalara yol açabilmektedir. Çalışır haldeki bir modülün çıkış parametrelerinin elde edilmesi için modül çalışırken yarıiletkenlerin parametrelerinin bulunması gerekir. Ancak çalışan bir modülün içindeki yarıiletkenlerin parametrelerinin bulunması çok zordur.

2.3.3. Klasik yöntem

Klasik hale gelmiş ölçüm ve test yönteminin temelini;

$$Q_C = \alpha \cdot I \cdot T_C - 0,5 \cdot I^2 \cdot R - K \cdot (T_H - T_C) - Q_L$$

$$Q_H = \alpha \cdot I \cdot T_H + 0,5 \cdot I^2 \cdot R - K \cdot (T_H - T_C)$$

$$R = \left(\frac{1}{\sigma_n} + \frac{1}{\sigma_p} \right) \cdot \frac{h}{a}$$

$$K = (\lambda_n + \lambda_p) \cdot \frac{a}{h}$$

(2.7)

$$P = I^2 \cdot R + \alpha \cdot (T_H - T_C) \cdot I$$

$$COP = \frac{Q_C}{P}$$

$$Z = \frac{\alpha^2}{R \cdot K}$$

eşitlikleri oluşturmaktadır. Burada Q_C modülün soğuyan yüzeyinden soğurulan toplam ısı yükü (Watt), Q_H modülün ısınan yüzeyinde açığa çıkan toplam ısı miktarı (Watt), modüle uygulanan akım şiddeti (A), R termoelementin direnci (Ω), K termoelementin ısı iletkenlik katsayısı ($W \cdot K^{-1}$); α_n , σ_n , λ_n n tipi yarıiletkenin α_p , σ_p , λ_p ise p tipi yarıiletkenin seebeck, iletkenlik, ısı iletkenliği katsayısı, h yarıiletkenin yüksekliği, a kesit alanı, $\alpha = (|\alpha_p + \alpha_n|)$ ise termoelementin toplam seebeck katsayısı; $\Delta T = T_H - T_C$ yüzeyler arasındaki sıcaklık farkı, T_C soğuyan yüzeyin sıcaklığı, T_H ısınan yüzeyin sıcaklığı, $Q_L = Q_{rad} + Q_{conv}$ watt olarak alınan toplam dış ısı yükü, Q_{rad} watt olarak toplam radyasyon dış ısı yükü, Q_{conv} watt olarak toplam konveksiyon dış ısı yükü, P TE modülün tükettiği güç (Watt), COP TE

modülün ısı performans katsayısı ve son olarak Z ise kalite katsayısı olarak tanımlanmaktadır [1-8].

Elde edilen bu denklemler bir TE modül ya da sistemin ısı parametrelerinin hesaplanmasının temelini oluşturmaktadır. Bu formülleri kullanarak yapılan hesaplamalar çok büyük hatalara yol açmaktadır. Çünkü elde edilen parametreler çalışır haldeki bir modülün gerçek şartlardaki parametreleri değildir. Çalışan bir modülün çıkış parametrelerinin bu yöntemle değerlendirilmesi daha abartılı olup gerçek değerlerden farklıdır. Bunun nedeni α , ρ , κ ve z gibi parametrelerin sadece sıcaklığa göre değil yarıiletkenin yüksekliğine göre de değişebilmesidir. Buna ilaveten bu formüllerde gerçek bir modülün yapısal ve teknolojik faktörlerinin hesaba katılmamış olması kullanılan yöntemi yetersiz kılmaktadır [31,32].

2.3.4. İdealize model

Melcor tarafından geliştirilen ve halen kullanılmakta olan idealize modelin temelini Eş. 2.8 oluşturmaktadır ve bu yöntemde α , ρ ve κ katsayılarının sıcaklığa göre değişimi hesaba katılmaktadır [24].

$$\begin{aligned}
 Q_C &= 2N \left(\alpha I T_C - \frac{I^2 \rho}{2G} - k \Delta T G \right) \\
 Q_H &= 2N \left(\alpha I T_H + \frac{I^2 \rho}{2G} - k \Delta T G \right) \\
 P &= 2N \left(\frac{I^2 \rho}{G} + \alpha I \Delta T \right) \\
 I_{\max} &= \frac{kG}{\alpha (\sqrt{1 + 2zT_H} - 1)} \\
 V_{\max} &= 2N \left(\frac{I\rho}{G} + \alpha \Delta T \right) \\
 \Delta T_{\max} &= T_H - \frac{\sqrt{1 + 2zT_H} - 1}{z}
 \end{aligned} \tag{2.8}$$

$$\eta = \frac{T}{\Delta T} \left(\frac{\sqrt{1 + 2zT} - 1}{\sqrt{1 + 2zT} + 1} \right)^{-\frac{1}{2}} = \text{COP}$$

Burada $G = a / h$ termoelementin geometrik faktörü, N ise TE modülü oluşturan termoelement sayısını ifade etmektedir. $\alpha = \alpha_n = \alpha_p$, $\rho = \rho_n = \rho_p$, $k = k_n = k_p$, $z = z_n = z_p$ ve $T = T_{\text{avg}} = (T_H + T_C)/2$ 'dir. Burada T bir termoelement kolunun (sütun) ortalama sıcaklığıdır. TE yarıiletkenin α , ρ , κ ve z katsayılarının ortalama sıcaklığa bağlı değişim fonksiyonları ise [24];

$$\begin{aligned} \alpha &= [\alpha_0 + \alpha_1 T + \alpha_2 (T)^2] 10^{-9} \quad \frac{V}{K} \\ \alpha_0 &= 2224; \quad \alpha_1 = 930,6; \quad \alpha_2 = -0,9905 \\ \rho &= [\rho_0 + \rho_1 T + \rho_2 (T)^2] 10^{-8} \quad \Omega \text{cm} \\ \rho_0 &= 5112; \quad \rho_1 = 63,4; \quad \rho_2 = 0.6279 \\ k &= [k_0 + k_1 T + k_2 (T)^2] 10^{-6} \quad \frac{W}{\text{cm K}} \\ k_0 &= 62605; \quad k_1 = -277,7; \quad k_2 = 0.4131 \\ z &= \frac{\alpha^2}{\rho k} \\ R &= \frac{2\rho N}{G} \end{aligned} \tag{2.9}$$

şeklinde ifade edilmektedir.

2.4. Yeni Yöntem

TE modüllerin çıkış parametrelerinin elde edilmesine yönelik olarak; AHISKA tarafından geliştirilmiş olan yeni yöntemin temelini, geleneksel yöntemler için Eş. 2.7'de kullanılan denklemlerin eşdeğer eşitliklere uğratılması ve sadece çalışan modülün çektiği akım (I), uçlarında düşen gerilim (V), ürettiği termoemk (E) ve herhangi bir yüzeyindeki sıcaklık (T_H ya da T_C) değerinin ölçülmesi oluşturmaktadır. TE modülün performansını etkileyecek iç ve dış bütün faktörlerin toplam etkileri deneysel parametre olarak adlandırılan V_{max} , I_{max} ve E_{max} değerlerinin kullanılmasına dayanan yeni yöntemle otomatik

olarak hesaba katılmaktadır. Çünkü deneysel parametreler TE modülün çalışması esnasında doğrudan ölçülerek elde edilmektedir [33].

Bilindiği gibi yüksüz çalışan bir ideal TE modül için; toplam ısı yük $Q_L = 0$ iken, birim zamanda üretilen soğuk enerji miktarı $Q_C = 0$ olmaktadır. Bu durumda soğuyan yüzeyin sıcaklığı T_{Cmin} , yüzeyler arasındaki sıcaklık farkı ΔT_{max} olmaktadır. Aynı zamanda TE modülün çektiği akım I_{max} , uçlarında düşen gerilim V_{max} ve ürettiği termoemk değeri de E_{max} olarak ifade edilmektedir.

TE modül uçlarında düşen maksimum gerilim;

$$V_{max} = \alpha \cdot T_H \quad (2.10)$$

$$V_{max} = I_{max} \cdot R + \alpha \cdot \Delta T_{max} = I_{max} \cdot R + E_{max}$$

olmak üzere iki şekilde ifade edilebilmektedir. Burada;

$$\alpha \cdot \Delta T_{max} = \alpha \cdot (T_H - T_{Cmin}) = E_{max} \quad (2.11)$$

olur. Böylelikle, TE modülün elektriksel direnci (R) ohm olarak Eş. 2.12 ile ifade edilebilir.

$$R = \frac{V_{max} - E_{max}}{I_{max}} \quad (2.12)$$

Bu şartlar için ısı denge denklemi, Eş. 2.13'deki gibi yazılabilir.

$$\alpha \cdot I_{max} \cdot T_{Cmin} - 0,5 \cdot I_{max}^2 - K \cdot \Delta T_{max} = 0 \quad (2.13)$$

Ayrıca Eş. 2.12 kullanılarak,

$$T_{Cmin} = T_H - \frac{E_{max}}{\alpha} \quad (2.14)$$

olarak bulunur. Elde edilen Eş. 2.12 ve Eş. 2.14 ifadeleri Eş. 2.13'de yerine konulursa,

$$(V_{\max} - E_{\max}) \cdot I_{\max} - 0,5 \cdot (V_{\max} - E_{\max}) \cdot I_{\max} = \left(\frac{K \cdot E_{\max}}{\alpha} \right) \quad (2.15)$$

denklemini bulunur. TE modülün termal iletkenlik katsayısı (K) değeri $W \cdot K^{-1}$ olarak;

$$K = \frac{0,5 \cdot \alpha \cdot (V_{\max} - E_{\max}) \cdot I_{\max}}{E_{\max}} = \frac{0,5 \cdot (V_{\max} - E_{\max}) \cdot I_{\max}}{T_H \cdot E_{\max}} \quad (2.16)$$

şeklinde yazılabilir. Elde edilen Eş. 2.12 ve Eş. 2.16, Eş. 2.7'de yerine yazılırsa ve $Q_L = 0$ kabul edilirse modülün Q_C değeri Eş. 2.17'deki gibi elde edilir.

$$Q_C = \alpha \cdot I \cdot T_C - \frac{0,5 \cdot I^2 \cdot (V_{\max} - E_{\max})}{I_{\max}} - \frac{\alpha \cdot \Delta T \cdot (V_{\max} - E_{\max}) \cdot I_{\max}}{E_{\max}} \quad (2.17)$$

TE modülün termoemk değeri Eş. 2.18 ile verildiğinden soğuyan yüzey sıcaklığı Eş. 2.19'teki gibi elde edilir.

$$E = \alpha \cdot \Delta T = \alpha \cdot (T_H - T_C) \quad (2.18)$$

$$T_C = T_H - \frac{E}{\alpha} \quad (2.19)$$

Eş. 2.17 ve Eş. 2.19'ten Q_C ,

$$Q_C = \alpha \cdot \left(T_H - \frac{E}{\alpha} \right) \cdot I - \frac{0,5 \cdot I^2 \cdot (V_{\max} - E_{\max})}{I_{\max}} - \frac{E \cdot (V_{\max} - E_{\max}) \cdot I_{\max}}{E_{\max}} \quad (2.20)$$

olarak bulunur. Bilindiği gibi modülün ısınan yüzeyinin sıcaklığı (T_H), bu yüzeyden ısıyı atan ısı transfer sistemine bağlıdır ve genelde sabit

tutulmaktadır. Çalışan gerçek bir TE modülün ısınan yüzeyinin sıcaklığı, daima yaklaşık olarak ısı transfer sisteminde kullanılan maddenin sıcaklığına eşittir. TE modüle uygulanan akım şiddeti (I) ile ısı yük (Q_C) değerindeki değişimler T_H 'nin değerini çok az etkilediği için, ilk yaklaşım olarak I_{max} için kullanılan Eş. 2.10, Eş. 2.20'te kullanılabilir. Bu durumda TE modülün soğuk yüzeyine gelen toplam ısı yükü (Q_C) watt olarak;

$$Q_C = V_{max} \cdot I - \frac{0,5 \cdot I^2 \cdot (V_{max} - E_{max})}{I_{max}} - \left(1 + \frac{0,5 \cdot (V_{max} - E_{max}) \cdot I_{max}}{E_{max}} \right) \cdot E \quad (2.21)$$

TE modülün ısınan yüzeyinde açığa çıkan toplam ısı (Q_H) watt olarak;

$$Q_H = V_{max} \cdot I + \frac{0,5 \cdot I^2 \cdot (V_{max} - E_{max})}{I_{max}} - \left(\frac{0,5 \cdot (V_{max} - E_{max}) \cdot I_{max}}{E_{max}} \right) \cdot E \quad (2.22)$$

şeklinde yazılabilir. Aynı şekilde TE modülün tükettiği güç (P) watt olarak Eş. 2.23 ile ifade edilir.

$$P = \frac{I^2 \cdot (V_{max} - E_{max})}{I_{max}} + E \cdot I \quad (2.23)$$

Bu durumda TE modülün verimi yani performans katsayısı (COP) Eş. 2.24'deki gibi elde edilir.

$$COP = \frac{Q_C}{P} \quad (2.24)$$

$$COP = \frac{V_{max} \cdot I - \frac{0,5 \cdot I^2 \cdot (V_{max} - E_{max})}{I_{max}} - \left(1 + \frac{0,5 \cdot (V_{max} - E_{max}) \cdot I_{max}}{E_{max}} \right) \cdot E}{\frac{I^2 \cdot (V_{max} - E_{max})}{I_{max}} + E \cdot I}$$

Bunlara ilaveten Eş. 2.10, Eş. 2.12 ve Eş. 2.16'a göre TE modülün kalite katsayısı (Z) değeri K^{-1} olarak;

$$Z = \frac{V_{\max} \cdot E_{\max}}{0,5 \cdot (V_{\max} - E_{\max})^2 \cdot T_H} \quad (2.25)$$

şeklinde yazılabilir. Bilindiği gibi bir TE modülün herhangi bir yüzeyindeki sıcaklığın ölçülmesi, özellikle modül bir sistemin parçası olup kapalı bir konumda bulunuyorsa çok zordur. Bunun için her bir yüzeyin üzerine termokuplun konulması ve çıkışlarının dışarıya çıkarılması gerekmektedir. Fakat TE modülün ısınan yüzeyi dışarıda olduğu için bu yüzeyin T_H sıcaklığı kolayca ölçülebilir.

Eş. 2.10 ve Eş. 2.14'e göre, TE modülün soğuyan yüzey sıcaklık (T_C) değeri K olarak;

$$T_C = T_H \left(1 - \frac{E}{V_{\max}} \right) \quad (2.26)$$

yazılabilir. Bu ifadeye göre T_C sıcaklığı doğrudan ölçülmeden, sadece T_H ve E değeri ölçülerek de bulunabilir. Aynı denklemden T_C ve E değerleri yardımıyla TE modülün ısınan yüzeyindeki sıcaklık (T_H) değeri K olarak;

$$T_H = \frac{T_C}{\left(1 - \frac{E}{V_{\max}} \right)} \quad (2.27)$$

TE modülün termoemk (E) değerinin farklı akımlardaki Q_C değerine göre değişimi ise volt olarak;

$$E = \frac{V_{\max} \cdot I - \frac{0,5 \cdot I^2 \cdot (V_{\max} - E_{\max})}{I_{\max}} - Q_C}{I + \frac{0,5 \cdot (V_{\max} - E_{\max}) \cdot I_{\max}}{E_{\max}}} \quad (2.28)$$

formülü ile ifade edilebilmektedir.

Elde edilen Eş. 2.12, Eş. 2.16, Eş. 2.17 ve Eş. 2.21 – Eş. 2.28 çalışma anında TE modüle ait tüm parametreleri vermektedir. Bu formüllere göre klasik yöntemlerde kullanılan TE yarıiletkenin α , ρ , κ ve z parametrelerin yerine I_{\max} , V_{\max} ve E_{\max} deneysel parametreler kullanılmıştır. Bu eşitliklerin kullanılabilmesi için hem her modül için farklı olan I_{\max} , V_{\max} ve E_{\max} deneysel parametre değerlerinin hem de T_C veya T_H sıcaklıklardan birinin doğrudan ölçülmesi gerekmektedir. Burada V_{\max} ve E_{\max} TE modülü oluşturan TE yarıiletkenleri karakterize ettiği için, bu parametreler yarıiletkenlerin geometrik özelliklerine bağlı değildir ve modülün makro büyüklüklerini oluşturmaktadırlar. Bu nicelikler çok büyük hassasiyetle kolayca ölçülebilirler. Elde edilen bu eşitlikler çalışan gerçek bir modülün parametrelerini karakterize etmektedirler. Ayrıca bu eşitlikler TE modül ve sistemlerin üretiminde kullanılan yapısal yöntemlerin ve malzemelerin kalitesinin bir göstergesi olarak da değerlendirilebilirler.

Elde edilen Eş. 2.21 – Eş. 2.26 çalışan bir termoelementin veya bir TE modülün çıkış ısı parametrelerini ifade etmektedir. Bu eşitliklerde yer alan $Q_C = Q_C(I, E)$, $Q_H = Q_H(I, E)$, $P = P(I, E)$, $COP = COP(I, E)$, $T_C = T_C(I, E)$ fonksiyonlarının analitik ifadeleri, uygulanan akım şiddetine ve termoelementin çalışma rejimine göre değişim göstermektedir. Bu durumları, $T_{C\min}$ ve $Q_{C\max}$ rejimlerinde olmak üzere uygulanan akım $I = I_{\max}$ ve $I < I_{\max}$ için 4 farklı şekilde incelenilebilir [39-40]:

$I = I_{\max}$ iken $T_C = T_{C\min}$ rejiminde; $Q_C = 0$ ve $E = E_{\max}$ olduğundan, TE modülün çıkış ısı parametreleri Eş. 2.29'teki gibi elde edilir.

$$Q_C = 0$$

$$Q_H = V_{\max} I_{\max}$$

$$P = V_{\max} I_{\max}$$

(2.29)

$$\text{COP} = 0$$

$$T_{C\min} = T_H \left(1 - \frac{E_{\max}}{V_{\max}} \right)$$

$$Z = \frac{V_{\max} E_{\max}}{0,5(V_{\max} - E_{\max})^2 T_H}$$

$I = I_{\max}$ iken $Q_{C\max}$ rejiminde; $T_C = T_H$ ve $\Delta T = 0$ yani $E = \alpha \Delta T = 0$ olduğundan, TE modülün çıkış ısı parametreleri Eş. 2.30'teki gibi elde edilir.

$$Q_{C\max} = 0,5 I_{\max} (V_{\max} + E_{\max})$$

$$Q_{H\max} = 0,5 I_{\max} (3V_{\max} - E_{\max})$$

$$P_{\min} = I_{\max} (V_{\max} - E_{\max})$$

(2.30)

$$\text{COP}_{\max} = 0,5 + \frac{E_{\max}}{V_{\max} - E_{\max}} = \frac{V_{\max}}{V_{\max} - E_{\max}} - 0,5$$

$$T_C = T_H$$

$$Z = \frac{V_{\max} E_{\max}}{0,5(V_{\max} - E_{\max})^2 T_H}$$

$I < I_{\max}$ iken $T_C = T_{C\min}$ rejiminde, TE modülün çıkış ısı parametreleri Eş. 2.31'teki gibi elde edilir. Burada I_{\max} 'tan küçük her bir akım için modül $T_{C\min}$ sağladığında üreteceği maksimum bir emk E_{\max}^I olarak gösterilmiştir.

$$Q_C = 0$$

$$Q_H = V_{\max} I + \frac{0,5 I^2 (V_{\max} - E_{\max})}{I_{\max}} - \frac{0,5 (V_{\max} - E_{\max}) I_{\max} E_{\max}^I}{E_{\max}} \quad (2.31)$$

$$P = \frac{I^2(V_{\max} - E_{\max})}{I_{\max}} + IE_{\max}^I$$

$$\text{COP} = 0$$

$$T_{C \min}^I = T_H \left(1 - \frac{E_{\max}^I}{V_{\max}} \right)$$

(2.31)

$$Z = \frac{V_{\max} E_{\max}}{0,5(V_{\max} - E_{\max})^2 T_H} = \frac{E_{\max} E_{\max}^I}{0,5(V_{\max} - E_{\max})^2 \Delta T}$$

$$E_{\max}^I = \frac{V_{\max} I - \frac{0,5I^2(V_{\max} - E_{\max})}{I_{\max}}}{I + \frac{0,5(V_{\max} - E_{\max})I_{\max}}{E_{\max}}}$$

$I < I_{\max}$ iken $Q_{C \max}$ rejimde; TE modülün çıkış ısıl parametreleri Eş. 2.32'deki gibi elde edilir.

$$Q_{C \max}^I = V_{\max} I - \frac{0,5I^2(V_{\max} - E_{\max})}{I_{\max}}$$

$$Q_H = V_{\max} I + \frac{0,5I^2(V_{\max} - E_{\max})}{I_{\max}}$$

2.32)

$$P = \frac{I^2(V_{\max} - E_{\max})}{I_{\max}}$$

$$\text{COP} = \frac{V_{\max} I_{\max}}{I(V_{\max} - E_{\max})} - 0,5$$

(2.32)

$$T_C = T_H$$

$$Z = \frac{V_{\max} E_{\max}}{0,5(V_{\max} - E_{\max})^2 T_H}$$

Q_{Cmax} rejiminde; $I < I_{max}$ iken elde edilen COP değeri, $I = I_{max}$ iken elde edilenden daha büyüktür ve teoride $I \rightarrow 0$ gittiğinde $COP \rightarrow \infty$ gitmektedir. Ayrıca yine Q_{Cmax} rejiminde her bir akım için bir Q_{Cmax}^I değeri söz konusudur. Eş. 2.29 – Eş. 2.32 incelendiğinde ise Z değerinin TE modülün çalışma rejimlerine bakılmaksızın hep aynı kaldığı görülmektedir.

TE yarıiletkenin α , ρ , κ ve z parametrelerinin, yeni yöntemin temelini oluşturan V_{max} , I_{max} ve E_{max} deneysel parametrelerine bağlı olarak basit bir şekilde ölçülebilmesi mümkündür. Bu doğrultuda Eş. 2.28 ve Eş. 2.29'teki ifadelerden yararlanılarak;

$$V_{min} = \frac{2NI_{max}\rho}{G} \quad (2.33)$$

$$E_{max} = 2N\alpha\Delta T_{max} \quad (2.34)$$

olarak elde edilmiştir. Ayrıca TE yarıiletkenin Seebeck katsayısı (α) V/K olarak;

$$\alpha = \frac{V_{max}}{2 \cdot N \cdot T_H} \quad (2.35)$$

TE yarıiletkenin termal iletkenliği (κ) W/(Cm.K) olarak;

$$\kappa = \frac{0,5I_{max}V_{min}V_{max}}{2 \cdot N \cdot G \cdot E_{max} \cdot T_H} \quad (2.36)$$

TE yarıiletkenin kalite katsayısı (z) K^{-1} olarak;

$$z = \frac{E_{max} \cdot V_{max}}{0,5 \cdot V_{min}^2 \cdot T_H} \quad (2.37)$$

TE yarıiletkenin elektriksel iletkenliği (ρ) $\Omega.cm$ olarak;

$$\rho = \frac{GV_{min}}{2 \cdot N \cdot I_{max}} \quad (2.38)$$

şeklinde ifade edilebilmektedir. Burada TE modülü karakterize eden V_{max} , I_{max} ve E_{max} deneysel parametrelerin doğrudan yarıiletkenin durumunu belirleyen α , ρ , κ ve z parametrelerinin fonksiyonel özelliklerine bağlı olduğu görülmektedir.

3. YAPAY ZEKÂ VE GENETİK ALGORİTMA

Yapay Zekâ (YZ); zekâ ve düşünme gerektiren işlemlerin bilgisayarlar tarafından yapılmasını sağlayacak araştırmaların yapılması ve yeni yöntemlerin geliştirilmesi hususunda çalışılan bilim dalıdır. YZ; “düşünme, anlama, kavrama, yorumlama ve öğrenme yapılarının programlamayla taklit edilerek problemlerin çözümüne uygulanması” olarak da ifade edilebilir. Daha geniş anlamda YZ:

- “Bilgisayarların, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekâsına özgü kapasitelerle donatılması bilimi”

veya

- “İnsanların düşünüş sürecini, çeşitli makinalar kullanarak taklit etme”

veya

- “İnsanın düşünme yapısını anlama, bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışma,”

veya

- “Programlanmış bir bilgisayarın düşünme girişimi”

olarak da tarif edilmektedir. YZ tanımının, zekâ tanımından olduğu gibi kolayca tarif edilemeyeceği ortadadır. Farklı bilim adamlarına göre tanımlarda değişmektedir.

Yapay zeka teknikleri

YZ teknikleri, yapay sinir ağları, bulanık mantık, sezgisel algoritmalar (genetik algoritmalar, tabu arama, karınca algoritması, ısıt işlemler, bağışıklık sistem algoritması gibi) ve uzman sistemler olmak üzere gruplandırılırlar [41,42].

3.1. Optimizasyon Problemlerinin Çözüm Teknikleri ve Genetik Algoritma

Optimizasyon problemlerinin çeşitli sınıflandırılması yapılmıştır. Ancak genel kabul gören sınıflandırmaya göre [16];

- Amaç fonksiyonunun, parametrelerle ilgili herhangi bir sınırlama olmaksızın minimizasyonu veya maksimizasyonu, *sınırlamasız optimizasyon*,
- Parametrelerle ilgili sınırlamanın veya sınırlamaların olduğu optimizasyon problemi ise *sınırlamalı optimizasyon*,
- Amaç fonksiyonunun ve parametrelerin lineer olup olmamalarına göre, bir optimizasyon problemi *lineer optimizasyon*,
- Amaç fonksiyonu veya parametreler nonlineer ise *nonlineer optimizasyon*,
- Tasarım değişkenlerinin alacağı değerler sürekli değerler ise bu tür problemlere *sürekli optimizasyon* problemi denir.

Optimizasyon problemlerinin çözümünde farklı teknikler kullanılmaktadır. Bu teknikler çeşitli kriterlere göre sınıflandırılabilir. Çizelge 3.1'de optimizasyon tekniklerinin sınıflandırılması gösterilmiştir [43-44].

Çizelge 3.1. Optimizasyon tekniklerine göre örnekler

Matematiksel Programlama Teknikleri	Stokastik Süreç Teknikleri	Yapay Zeka Optimizasyon Teknikleri
Klasik Analiz	İstatistiksel Karar Teorisi	Genetik Algoritma
Doğrusal Olmayan Programlama	Markow Süreçleri	Tavlama Benzetim
Doğrusal Programlama	Yenilenme Teorisi	Tabu Araştırması
Dinamik Programlama	Benzetim Metodu	Karınca Koloni Algoritması
Oyunlar Teorisi	Güvenilirlik Teorisi	Diferansiyel Gelişim Algoritması
Kareli Programlama		Yapay Bağışıklık Algoritması

Yapay Zeka Optimizasyon Teknikler: optimizasyon problemlerinin çözümüne yönelik yapay zeka optimizasyon teknikleri meta sezgisel araştırma teknikleri olarak da bilinmektedir. Bu algoritmalar herhangi bir çözümü gerçekleştirmek veya hedefe varmak için çeşitli alternatif hareketlerden etkili olanlara karar vermek amacı ile tanımlanan kriterler veya bilgisayar metotlarıdır. Kesin çözüm yöntemleri ile çözüm sağlanabilen problem büyüklüklerinin kısıtlı olması ve ihtiyaç duyulan hesap sürelerinin büyüklüğü araştırmacıları daha büyük problemler için en iyi çözüm olmasa da “iyi” bir çözüme süratle ulaşan metotlar üzerinde çalışmaya yöneltmiştir. Yapay zeka optimizasyon algoritmalarının uygulamalarının uygulama sonuçları aynı zamanda kesin çözüm metotları için bir alt veya üst sınır oluşturmaları nedeni ile de önemlidir. Bu tür çözüm algoritmaları kesin çözümü garanti etmezler ancak amaca uygun bir çözümü bulmayı garanti edebilirler.

Uygulama sonuçları göstermektedir ki yapay zeka optimizasyon algoritmaları probleme bağımlı algoritmalarlardır. Yani bir problemde başarılı olurken diğer bir problem için aynı şekilde başarılı olmayabilirler. Yapay zeka optimizasyon algoritmaların başlıcaları; yapay ısı işlem algoritmaları, genetik algoritmalar, tabu araştırma algoritmaları sayılabilir [45].

Yapay Isıl İşlem algoritması: Bu yaklaşım, metalik malzemelerin katı hal sıcaklık değişmelerinin birbirine bağlı birkaç işlemle, amaca uygun özellik değişmelerinin sağlanması olarak tarif edilir. Kirkpatrick ve ark. (1983) tarafından geliştirilmiş olan bu yaklaşım, metallerin ısı işlemi ile bir optimizasyon probleminin çözümü arasındaki benzerlikleri temel almaktadır. Isıl işlem, bir katının sıcaklığının belirli bir maksimum dereceye kadar artırılarak tekrar eritilmesi şeklinde tanımlanır.

Genetik algoritma: Genetik algoritmalar Holland tarafından biyolojik evrim sürecinden hareketle önerilmiştir. Burada tanıtılan diğer iki metottan farklı olarak, genetik algoritma tek bir çözüm yerine aynı anda birden fazla sayıda çözümü değerlendirir. Bu özellik, genetik algoritmanın paralel arama özelliği olarak nitelendirilir. Problem çözümü genlerden oluşan bir kromozom ile ifade edilir [18].

Tabu Arama: Glover tarafından geliştirilmiş olan bu algoritma, önceleri ayrık optimizasyon problemlerine uygulanmıştır. Bu yaklaşımda bütün olasılıkları denemek çok fazla zaman alacağından dolayı zekice bir yaklaşımla, en az hesaplama ile en iyi çözüme ulaşmak için tabu arama (TA) algoritması geliştirilmiştir.

TA algoritmasında; çözüm kümesi, komşuluk ve tabu listeleme işlemleri önemli unsurlardır. Çözüm kümesi 1 ve 0'lardan oluşan bir vektörle ifade edilir. En uygun olarak kullanılan komşu üretme yaklaşımları bit değişimi ve komşu çözümlerin ters dizilimidir [41].

Karınca Algoritması: Bu algoritma, matematiksel olarak karınca koloni davranışlarının modellenmesine dayalı bir yaklaşımdır. İlk çalışma, Dorigo ve arkadaşları tarafından 1991 yılında yapılmıştır. Bu araştırmacılar geliştirdikleri sisteme karınca sistemi, ortaya çıkan algoritmaya ise karınca algoritması ismini vermişlerdir [41].

3.2. Genetik Algoritmalar

Genetik Algoritmanın Tarihçesi

Genetik algoritmalar ilk olarak John Holland tarafından ortaya atılmıştır. John Holland, 1962 yılından başlamak üzere doğanın seçim kriterleri ve doğaya uygunluk ile ilgili dersler vermiştir. John Holland, Charles Darwin' in "en iyinin hayatta kalması" prensibinden yola çıkarak, 1970'li yılların başında genetik algoritmalar ile ilgili çalışmalarına başlamıştır [18,19] .

Genetik algoritmalar, prensiplerini evrim teorisinden alan, stokastik uygulamalarda ve optimizasyon konularında oldukça geniş bir uygulama alanına sahip bir yöntemdir. Stokastik optimizasyon tekniklerinde doğal seleksiyon işleminin uygulama aşamasında evrim algoritmaları arasında; genetik algoritmalar, evrimsel programlama ve evrim stratejileri yer almaktadır. Bugün evrimsel algoritmalar arasında en çok bilinen ve kullanılan ise genetik algoritmalarıdır. Genetik algoritmalar gerçek hayata ilişkin problemlerin çözümünde geleneksel optimizasyon problemlerine oranla daha başarılı sonuçlar üretirler [46].

Genetik Algoritmanın Temelleri

Genetik algoritmanın genel formu Goldberg (1989) tarafından belirtilmiştir. Genetik algoritma, stokastik arama tekniğinin doğal seleksiyon ve genler ile işletilmesi mekanizmasına dayanır. Genetik algoritmanın diğer geleneksel

arama tekniklerinden farkı, popülasyon adı verilen rastgele seçilmiş başlangıç çözümünden hareketle algoritmanın uygulanmasıdır [49].

Popülasyondaki her bir bireye kromozom denilmektedir. Kromozomlar dizini sembolü olarak kullanılırlar. Kromozomlar kural olmamakla birlikte kullanım aşamasında ikili sayı düzeninde bit dizini olarak ifade edilirler. Kromozomların birbirini takip eden iterasyonlarla evrim geçirmelerine jenerasyon adı verilir. Her bir jenerasyon süresince kromozomlar uygunluk koşulu uyarınca evrim geçirirler. Yeni jenerasyon yaratmak için, mevcut jenerasyonun iki kromozomu çaprazlama operatörü ve mutasyon operatörü kullanılarak elde edilir. Elde edilen yeni bireye döl denir. Yeni jenerasyonda uygunluk kriterine uyan ebeveynler ve dölleri, jenerasyon içerisinde yer alırken, uygunluk koşuluna uymayan ebeveynler ve dölleri jenerasyondan yok edilirler. Bu işlem doğal seleksiyon işlemi adını alır. Böylelikle en iyilerin yaşama şansının bulunduğu jenerasyonların popülasyon büyüklüğü korunmuş olur. Bu işlem uyarınca en uygun kromozomların seçilme şansı maksimize edilmiş olur. Birkaç jenerasyon sonra, algoritma en iyi kromozomların bulunduğu optimum çözüme yakınsar.

Genetik algoritma, klasik optimizasyon metodlarından daha etkili ve çözüm yaklaşımında yapılacak küçük değişikliklerle ilerlemesinden dolayı da daha esneklerdir. Bu sebeple, genetik algoritma araştırmacıların ilgisini çekmektedir. Bilindiği üzere optimizasyondaki temel amaç optimal bir noktaya ulaşabilmek, daha doğrusu mümkün oldukça yaklaşımdır. Bunu gerçekleştirmek için bilinen pek çok klasik yöntem vardır. Bu yöntemlerin başarısı optimal noktaya ulaşmış olup olmadığı veya ne kadar ulaşabildikleri ile ölçülür. Genetik algoritmalar, klasik algoritmalarından dört temel noktada ayrılır:

(i) Genetik algoritma, parametrelerin kendileri ile değil, parametre takımının kodlanmış bir haliyle uğraşırlar.

(ii) Genetik algoritma, aramaya tek bir noktada değil, bir nokta ailesinden başlarlar. Dolayısıyla yerel bir optimuma takılmadan çalışabilirler.

(iii) Genetik algoritma, amaç fonksiyonunun türevlerini ve bir takım ek bilgileri değil, doğrudan amaç fonksiyonunun kendisini kullanırlar.

(iv) Genetik algoritmada deterministik değil rastlantısal geçiş kuralları kullanılır [19].

3.3. Genetik Algoritmanın Temel Yapısı

Genetik Algoritma ile problemlerin çözümünde aşağıda belirtilen adımlarla şu şekilde bir süreç söz konusudur.

Adım 1. Genetik Algoritma sürecinin işletiminden önce problemin doğasına uygun olarak kodlama yapılır.

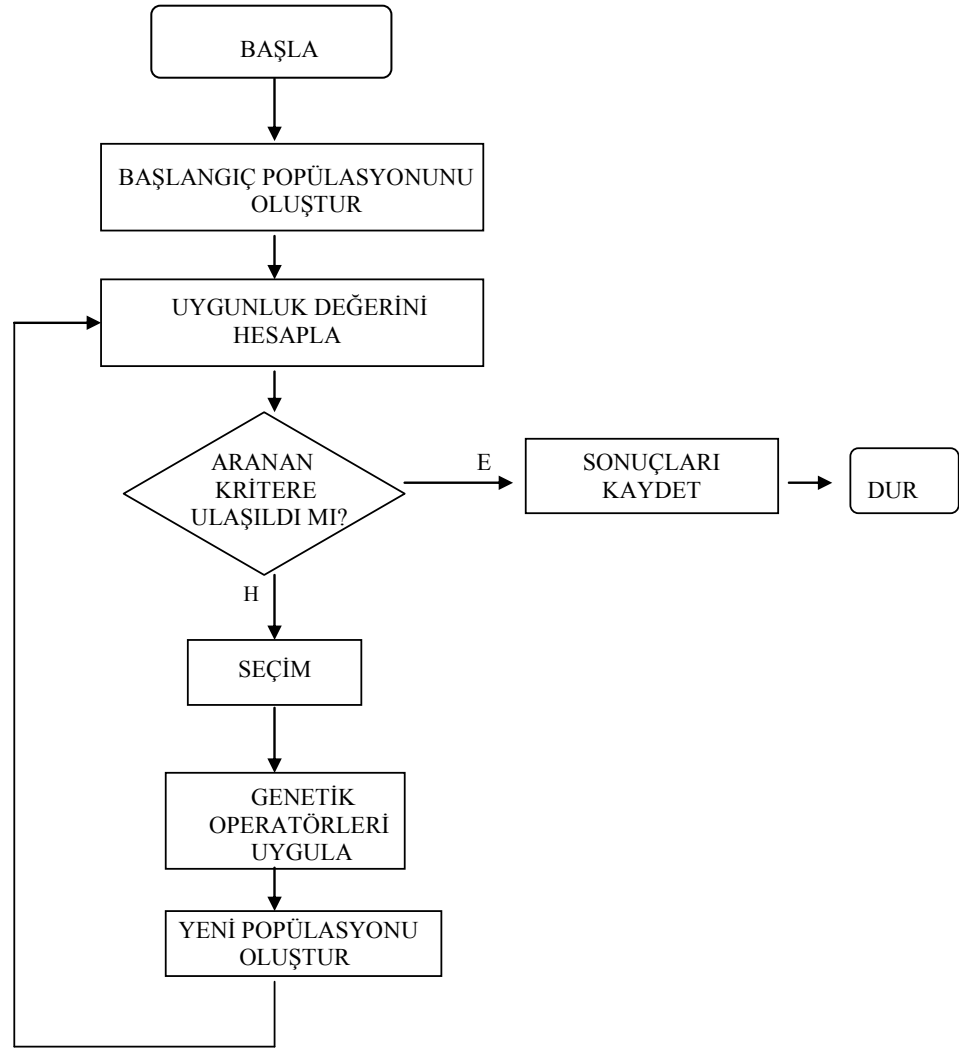
Adım 2. Genellikle rastgele olarak başlangıç popülasyonu yaratılır.

Adım 3. Başlangıç popülasyonunda yer alan her bir dizinin uyum değeri hesaplanır.

Adım 4. Popülasyonu değiştirmek ve yeni jenerasyonlar yaratmak için ise seçme, çaprazlama ve mutasyon operatörleri kullanılır.

Adım 5. Yeni popülasyona uyum değeri atanarak en iyi çözüm değeri bulunana kadar genetik algoritma operasyonları devam ettirilir.

Bir problemin çözümü için kullanılan genetik algoritmanın akış şeması şekil 3.1. de verilmiştir [48-50].



Şekil 3.1.Genetik algoritmanın akış şeması

3.3.1. Kromozomların temsili (kodlama)

Genetik algoritmanın uygulanabilmesi için yapılması gereken ilk adım, bağımsız parametrelerin (genlerin) kromozomlar içerisinde kodlanmasıdır. Kromozomlar problemin çözümü için bilgi taşıyan yapılardır. Kodlama ile ilgili çeşitli yöntemler olmasına karşın, en yaygın kullanılan kodlama yapısı ikilik düzende yapılan kodlamalardır. Goldberg (1989)' in temellerini attığı basit genetik algoritma ikili dizilerden oluşan bir popülasyon üzerinde çalışır. Burada kromozomlar 0 ve 1'lerden oluşan dizilerdir. Her parametre için "n" bit kullanılır. "n", her parametre için farklı olabilir [51]. Kromozomların kodlanmasında, tamsayı dizisi ya da kurallar dizisi olarak çeşitli gösterim

biçimleri vardır. Genetik algoritmanın her işletilmesinde, popülasyondaki dizilerin bir değerlendirme fonksiyonu yardımıyla uygunluk değeri hesaplanır. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirmekte ve bu parametrelere göre hesaplama yapılmaktadır. Genellikle genetik algoritmaların başarısı bu fonksiyon verimli ve hassas olmasına bağlıdır. Genetik algortmada bağımsız parametrelerin kromozomlar içinde kodlanması gerekmektedir [52].

3.3.2. Başlangıç popülasyonunun oluşturulması

Genetik algoritmanın uygulamasında ilk adım, kromozomların bir başlangıç popülasyonunun oluşturulmasıdır. Buna göre rastgelelik ilkesine dayalı başlangıç popülasyonunun oluşturulması yaygınlıkla kullanılan bir metottur. Genetik algoritma başlangıç popülasyonundan hareketle algoritma adımlarını kullanarak optimum çözüme her iterasyonda yakınsayan yeni popülasyonların (jenerasyon) oluşturulmasını sağlar. Genetik algoritma sürecinde başlangıçta belirlenen popülasyon büyüklüğünün (n) yüksek seçilmesi durumunda çözüm uzayı iyi örnekleneceğinden arama etkinliği sağlanabilecektir. Fakat bu hızlı bir şekilde yüksek çözüm kalitesine erişmede aynı performansı gösteremez. Popülasyon büyüklüğünün yeterince büyük seçilemediği durumlarda ise optimal çözüm uzayının ortaya çıkmaması problemiyle karşılaşılabilir. Yapılan çalışmalarda önerilen popülasyon büyüklüğü 100-300 arasındadır [53].

3.3.3. Amaç fonksiyon veya uyum değeri

Genetik Algoritma ile problem çözümünde probleme uygun çözüm kümesinin elde edilebilmesi için uyum fonksiyonunun belirlenmesi gerekmektedir. Uyum fonksiyonunun genetik algoritma yapısı içerisinde kullanılması ile problemin çözümünde yarar sağlayacak olan kromozomların yaşama şansının artırılması, problemin çözümünde yarar sağlamayan kromozomların ise

popülasyon içerisinde ayıklanmasına olanak sağlanır (Doğal Seleksiyon İlkesi). Böylece algoritma, başarısız olan aday çözümlerini elimine eder. Bir optimizasyon probleminde maksimizasyon ya da minimizasyon için kullanılan fonksiyon, genetik algoritma içerisinde uyum fonksiyonu olarak değerlendirilir [54-55].

3.3.4. Seçim

Seçim operasyonu üremeye tabi olacak bireylerin seçilmesi olarak tanımlanabilir. Genetik algoritma'da seçme operatörünün işlevi, yüksek uyuma sahip bireylerin sonraki kuşaklarda var olma olasılığını artırmaktır. Böylelikle seçim, arama uzayının başarılı olacağı umulan bölgelerde yoğunlaştırılır [20]. Jenerasyonlarda oluşturulacak yeni bireylerin ebeveynlerinin nasıl seçileceğini ve kaçar tane yeni bireyin oluşturulacağı belirlenmelidir [21]. Bunun için üremeye tabi olacak bireyler, üreme havuzu olarak tabir edilen bir popülasyondan rastgele seçilerek eşleştirilir. Seçim için oluşturulan baskınlık, popülasyonun çeşitliliğinde oldukça büyük etkiye sahiptir. Eğer yüksek derecede seçim baskısı uygulanırsa, genetik algoritma, hızla popülasyon çeşitliliğini yitirerek yakınsama sürecine girer. Zayıf seçicilik uygulandığında ise, aramanın etkinliği olumsuz etkilenir [22].

Genetik algoritma yönteminde uyuma orantılı seçim yöntemleri kullanılır.

Uyuma orantılı seçiminde her i birey için p_i seçilme olasılığı;

$$P_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad (3.1)$$

Şeklinde bulunur. Burada f_i , i . Bireyin uyum değerini;

$\sum_{i=1}^n f_i$, popülasyonun toplam uyumunu göstermektedir. Uyuma orantılı seçim

planındaki f_i uyum değerine sahip bir bireye, f_i/\bar{f} adet yavru verilir [46].

Seçim mekanizması, problem çözücünün kararına bağlı olarak farklılık arz etmektedir. Holland (1975) bireyin bir önceki jenerasyondan rastgele seçilen bir birey ile değiştirilmesi ya da ebeveynlerini bir neslin tamamını değiştirecek şekilde seçme yoluna gitmiştir. Bu seçim yöntemleri uyarınca en iyinin bir sonraki jenerasyonda yer alması kesin değildir. Fakat De Jong (1975)'un elitist modeli uyarınca en iyinin mutlaka diğer jenerasyonlarda var olması sağlanmıştır [56].

Seçim mekanizması belirlendikten sonra uyum fonksiyonuna göre değerlendirme yapılır.

Rulet Tekerleği Yöntemi

Rulet tekerleği seçimi; yerine koyarak stokastik örnekleme olarak da adlandırılır. İlk defa Holland tarafından ortaya çıkarılmıştır. Öncelikle Eş. 3.1 yardımıyla her bir bireyin P_i seçilme olasılığı hesaplanır. Bu sayılar bir tabloda tutulur. Tablodaki olasılık değerleri birbirine eklenerek rastgele bir sayıya kadar ilerlenir. Bu rastgele sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu birey seçilmiş olur. Bu işlem N kez tekrarlanır. Bu yöntem rulet tekerleği ismi, bir daireyi, bireylerin uyum değerlerine orantılı olarak dilimleyip çevrildiğinde elde edilecek sonuçların benzeşim olduğu için verilmiştir [19].

Rulet tekerleğinde bir bireye pay edilen yavru sayısı, beklenen yavru sayısından önemli ölçüde farklı olabilir. İterasyon sayısı arttıkça hata olasılığı da artmaktadır. Pay edilen yavru sayısı, sadece çok büyük popülasyonlar için beklenen yavru sayısına yaklaşmaktadır [57]. Rulet tekerleği seçiminde bu örnekleme hatası nedeniyle arama, tahmin edilenden farklı yönlerde devam etmektedir. Bu ise algoritmanın yerel optimuma zamansız yakınsamasına

neden olmaktadır [56].

Hesaplanan uyum sonucunda önceden belirlenmiş seçim mekanizmalarından biri kullanılarak seçim yapılır. Seçme işlemi üreme havuzuna alınan bireylerin popülasyon büyüklüğüne ulaşmasıyla son bulur.

Turnuva Seçimi Yöntemi

Turnuva seçimi yöntemi, sıralama seçim yöntemine benzer. Ancak işlemsel olarak daha verimli olup, uygulamaya daha yakındır. Turnuva seçiminde, popülasyondan yerine koyarak ya da yerine koymadan rastgele "t" birey seçilir. "t" büyüklüğüne turnuva genişliği denir. Bu gruptaki en iyi birey yeni popülasyona kopyalanır. Bu işlem N kez tekrarlanır. Seçilen turnuva genişliğinin fazla olması, yöntemin seçicilik baskısını arttırır. Turnuvalarda çoğunlukla $t = 2$ alınır ve ikili turnuva olarak adlandırılır [22,53]. Yani rastgele iki birey seçilir ve seçilen iki bireyden uygunluk derecesi yüksek olan bir sonraki popülasyona aktarılır. İşlem popülasyondaki kromozom sayısı kadar tekrarlanır.

Elitist Seçim Yöntemi

İlk olarak De Jong tarafından önerilmiştir [21]. Bu yöntem, popülasyonun en iyi bir ya da iki bireyini koruyup, popülasyonun geri kalan elemanlarını uyuma orantılı seçim yöntemlerinden birini kullanarak yeni bireyler ile değiştirir. Böylece en iyi bireylerin yaşaması sağlanır [54]. Yöntemin avantajı en iyi uyum değerine sahip bireyin örnekleme hatasını ya da genetik operatörlerin kullanılması ile kaybolmasını önlemektir. Elitist yöntemi için aşağıdakileri söylenebilir:

- i. Belirli sayıdaki en iyi birey hiçbir işleme tabi tutulmadan doğrudan yeni jenerasyona aktarılır.
- ii. Elit birey sayısı arttıkça çözümdeki çeşitlilik azalır.

- iii. Elit birey alınmazsa yeni jenerasyonun en iyi bireyi bir önceki jenerasyonun en iyi bireyinden daha kötü olabilir.

3.3.5. Genetik operatörler

Seçim işlemi problemin çözümü ile ilgili bilgi verebilir. Ancak kesin bir çözüme götürmez. Seçim işlemi popülasyon büyüklüğüne ulaşıncaya kadar devam eder ve seçilen her birey, üreme havuzuna alınırlar. Seçilen bireyler yeni nasılın ebeveynleridir. Çözüme ulaşmayı sağlayacak yeni bireyler, bu aşamada genetik operatörler yardımıyla elde edilirler. Yeni çözümler yaratmada görev alan iki önemli genetik operatör vardır. Bunlar çaprazlama ve mutasyondur.

3.3.5.1. Çaprazlama

Genetik algoritmaya dayalı çoğu uygulamada en önemli operatör çaprazlamadır. Rastgele iki birey seçilir ve bireylerin çaprazlama noktaları da rastgele seçilir. Yeni elde edilen iki bireyin ikisi de yeni popülasyona aktarılabilir [59]. Literatürde en çok kullanılan çaprazlama operatörleri, tek noktalı çaprazlama, iki noktalı çaprazlama, çok nokta çaprazlama ve düzgün çaprazlamadır.

Tek nokta çaprazlama

Bu çaprazlama işlemi, L uzunluğundaki iki zincirin birer noktalarından ikiye bölünmesiyle birincil ve ikincil parçalarının yer değiştirmesi işlemidir. L uzunluğundaki bir zincir için L-1 tane çaprazlama noktası bulunabilmektedir. Rastgele seçilen çaprazlama noktalarının seçilme olasılığı birbirine eşittir.

Algoritmada belirlenen bir çaprazlama olasılığı (P_c) çaprazlama uygulanacak çözüm çiftleri belirlenir. Çoğunlukla 0,6 ile 1 arasında bir (P_c) olasılığı kullanılmaktadır [60]. Eğer $[0,1]$ aralığında rastgele seçilen bir sayı (P_c) den

küçük ise o çifte çaprazlama uygulanır. Diğer taraftan bu çözümler popülasyonda değişmeden kalır. Her yeni popülasyonda yaklaşık olarak $P_c L$ bireye çaprazlama uygulanır [60]. Sistemin işleyişinden de anlaşılacağı gibi her bir bireyin geninin mutlaka çaprazlanması gerekmez. Tek nokta çaprazlama Şekil 3.2'deki gibi gösterilebilir.

Çaprazlama için seçilen iki kromozon alındığında;

1101110010

1011001101

Öncelikle 1 ile n-1 tane arasında bir tamsayı rastgele seçilir (çaprazlama noktası). Bu nokta 3 olsun bu nokta Şekil 3.2 ile gösterilmiştir. 1. Ebeveyn ile 2. Ebeveyn çizgi çizildiği noktadan (3. Gen den sonra) yer değiştirerek yeni bireyler elde edilir.

1	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---

 Ebeveyn 1:

1	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

 Ebeveyn 2:

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

 Yeni Birey 1:

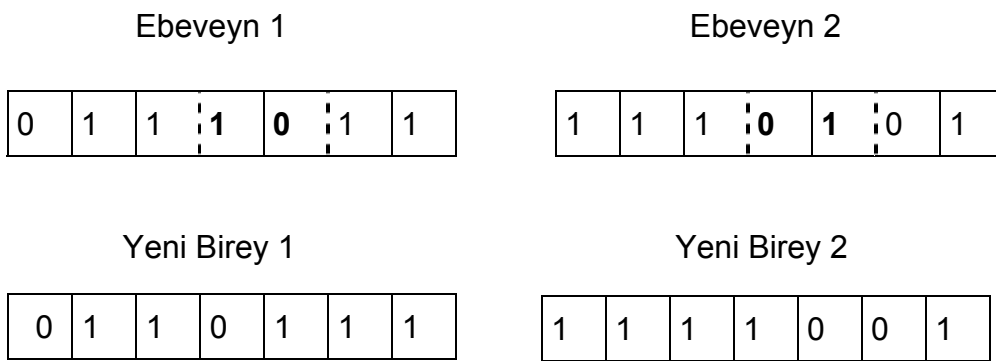
1	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

 Yeni Birey 2:

Şekil 3.2. Tek nokta çaprazlama

İki nokta çaprazlama:

İki nokta çaprazlamada, iki çaprazlama noktası vardır ve bu iki nokta arasında kalan alt dizilerin değiştirilmesiyle iki yeni birey elde edilir [59]. Şekil 3.3'de 0111011 ve 1110101 dizilerine İki nokta çaprazlamanın uygulanışı göstermektedir. Burada da çaprazlama noktaları, 1 ile n-1 arasında rastgele seçilmektedir.



Şekil 3.3. İki nokta çaprazlama

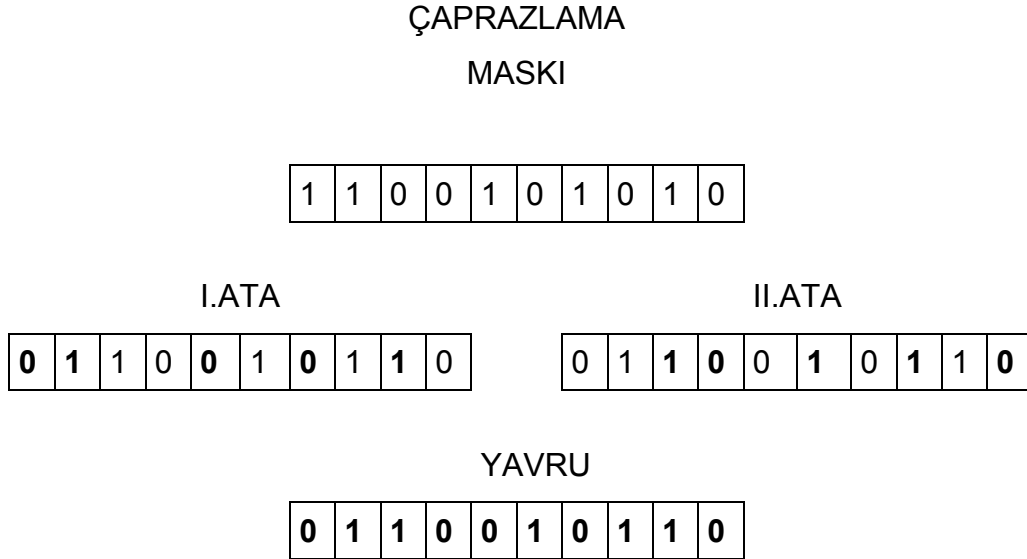
Çok nokta çaprazlama:

Çok nokta çaprazlamada, iki nokta çaprazlamanın bir uzantısıdır. Her bir çözüm, k çaprazlama noktası ile k parçaya ayrılır. Bir atlanarak elde edilen allel blokları çiftler değiştirilerek yavrular elde edilir [57].

Üniform (düzgün) çaprazlama

Diğer bir çaprazlama işlemi ise düzgün çaprazlamadır. Düzgün çaprazlama işlemi için öncelikle “Çaprazlama Maskı” tanımlanır. Çaprazlama maskesindeki 1, o genin 1. atadan kopyalanacağını, 0 ise o genin ikinci atadan kopyalanacağını ifade etmektedir. Çaprazlama maski tek noktada ya da iki noktada da kullanılabilir.

Rastgele yaratılan 1100101010 çaprazlama maskesinden hareketle düzgün çaprazlama işlemi şekil olarak Şekil 3.4'deki gibi gösterilebilir.



Şekil 3.4. Düzgün çaprazlama

Yapılan deneylerde çıkan sonuçlara göre; düzgün çaprazlama küçük popülasyonlar için, iki nokta çaprazlamanın ise, büyük popülasyonlar için daha iyi sonuçlar türettiği görülmektedir. Düzgün çaprazlama ile özellikle küçük popülasyonlarda oldukça keşifsel bir arama yapma imkanı sağlanmaktadır. Büyük popülasyonlarda var olan çeşitlilik, keşif gereksinimini azaltmakta ve iki nokta çaprazlamayı daha uygun kılmaktadır [58].

Düzgün çaprazlamada, zincirler daha çok parçalanacağı ve buna bağlı olarak zincirde çeşitlilik sağlanacağı için, problemde her hangi bir çaprazlama tekniğine karar verilemediği durumlarda seçim, düzgün çaprazlamadan yana yapılmalıdır.

Çaprazlama yapılırken en yüksek uyuma sahip bireyler seçildiği halde, çok modelli bir fonksiyon en iyileşmek istendiğinde farklı optimumdan iki zincir çaprazlandığında sonucun iyi olması garantisi yoktur [52].

3.3.5.2. Mutasyon

Genetik algoritmada bir diğer işlem mutasyondur. Herhangi bir kromozomdaki bir gen'in sebep olmadan rastgele değiştirilmesidir.

Mutasyona uğrayacak bireyleri tespit etmek için her bir bite $[0,1]$ aralığında rastgele bir olasılık verilir. Bu olasılık, problem tanımlanırken belirlenen mutasyon olasılığı P_m ile karşılaştırılır. P_m 'den büyük olasılığa sahip bireyler mutasyona uğramazlarken, P_m 'den küçük olasılığa sahip olan bireyler mutasyona uğrar ve değeri 0 olan bitler 1, 1 olan bitler ise 0 değerini alırlar.

Genellikle $P_m \in [0.001,001]$ çok küçük bir sayıdır. Böylece, çaprazlama işleminin işlevinin artması sağlanmış olur. Yani, P_m 'nin küçük olması, algoritmanın sonucunu büyük ölçüde algoritmaya bırakmak demektir [19,61].

Mutasyon kavramı Şekil 3.5'den daha iyi görülebilir.

Birey	1	0	1	1	0	1	1	0	1	1
Mutasyona Uğrayan Birey	1	0	1	1	1	1	1	0	1	1

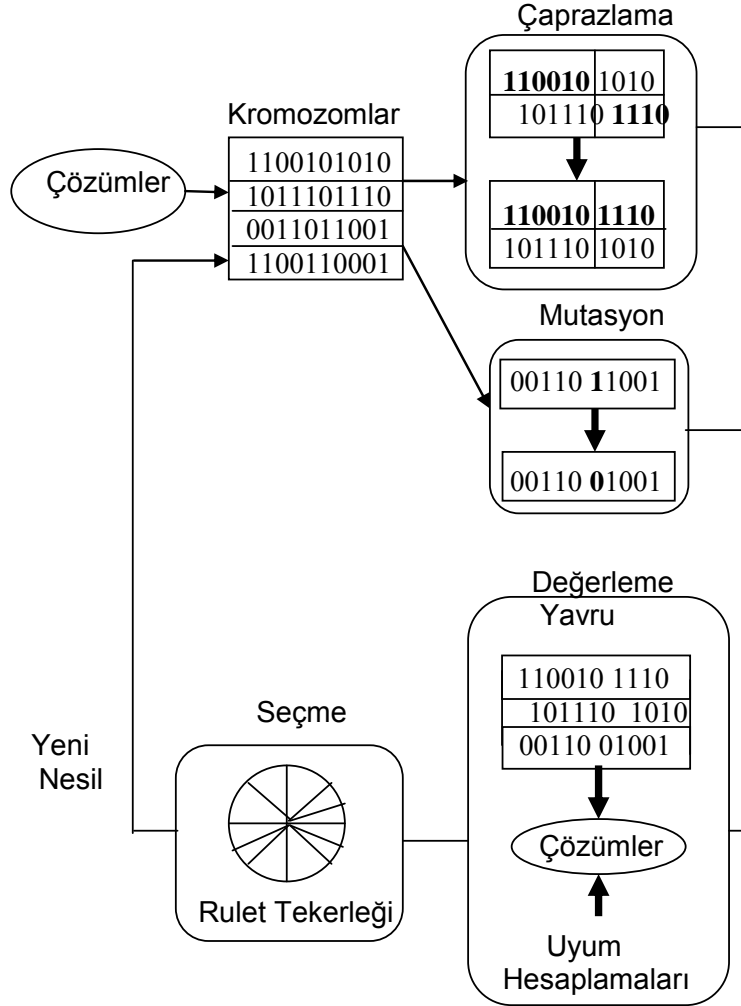
Şekil 3.5. Mutasyon

Şekil 3.5, beşinci genin mutasyona uğradığı durumu göstermektedir.

Mutasyon işlemi için bazı araştırmacılar tarafından önerilen genel kabul gören iki farklı parametre kümesi mevcuttur. Bunlar;

- Büyük popülasyon için genişlik değeri $n=100$, $P_c = 0,6$ ve $P_m = 0,001$
- Küçük popülasyon için genişlik değeri $n=100$, $P_c = 0,9$ ve $P_m = 0,01$ biçimindedir [59].

Şekil 3.6 Genetik algoritmaların genel yapısını göstermektedir [17].



Şekil 3.6. Genetik algoritmaların genel yapısı

3.4. Genetik Algoritmaların İşleyişi (Şema Teorisi)

GA'ların çalışması açıklamaya yönelik pek çok çalışma yapılmıştır. Bu çalışmalardan en çok bilineni şema (schema) kuramıdır. Holland GA'ların yüksek uyuma sahip bölgelerin nasıl bulunduğunu açıklanabilmek için şema kavramından faydalanılmıştır [61].

Genetik algoritmalarla oluşan başarılı bireyler incelenirse, bu bireyler arasındaki benzerlikler bulunabilir. Bu benzerliklerden yola çıkarak şemalar

oluşturulabilir. Şema, belirli pozisyonlarda benzerlikleri olan dizileri tanımlamaya yarayan bir benzerlik modelidir. Burada kullanılacak olan İkilik dizi kodlaması için {0, 1, *} alfabesini ele alalım. * sembolü, dikkate alınmayan ya da değeri değişken kart olarak adlandırılır. Böylece şema 0,1 ve * sembollerinden oluşan bir dizidir. * sembolü, hiçbir zaman GA tarafından işletilmez. Sadece, diziler arasındaki benzerliklerin ortaya konulabilmesini sağlayan gösterimsel bir araçtır. Beş biti olan dizileri ele alalım **000 şeması, son üç pozisyonunda 0 olan dizileri temsil eder.

Bunlar, 00000,10000,01000,11000 dizilerdir. Benzer biçimde, 1*00* şemasının, 10000,10001,11000 ve 11001 dizilerini temsil ettiği veya eşlediği söylenebilir. ***** şeması beş bite sahip tüm dizileri temsil eder. Bir şema tarafından temsil edilen her bir diziye o şemanın örneği denir. Genetik algoritma kendi içinde sanal olarak şemaları oluşturur. Genetik algoritmalar şemaları oluşturmak için toplum üyelerinin kodları dışında bir bilgi tutmaz. Bir şemanın sabit pozisyonları, 0 veya 1'lerin yer aldığı pozisyonlardır. 1*00* şemasında sabit pozisyonlar, 1., 3. ve 4. pozisyonlardır [59].

Bir H şemasının derecesi, sabit pozisyonların sayısıdır ve $o(H)$ ile gösterilir. (*1*0*1) şema yapısının derecesi, 3 tür. Derece, bir şemanın belirginliğini gösterir. Farklı derecelere sahip H şemaları şöyle verilebilir;

$$H_1=***000*111$$

$$H_2=**11*1****$$

$$H_3=11101**001$$

Şemaları, $O(H_1)=6$, $O(H_2)=3$, $O(H_3)=8$, derecelerine sahiptir. Burada en belirgin nitelikteki şema ise H_3 şemasıdır. Bir şemanın derecesi mutasyona rağmen yaşama olasılığını hesaplarken faydalı olmaktadır [64].

Şemanın tanımlayan uzunluğu, birinci ve sonuncu sabit pozisyonlar arasındaki uzaklıktır ve $l(H)$ ile gösterilir.

Örneğin $I(H_1)=10-4=6$, $I(H_2)=9-5=4$ ve $I(H_3)=10-1=9$ olur.

- 6 derecesine sahip H_1 şemasının tanımlayan uzunluğu, $I(H_1) = 10 - 4 = 6$,
- 3 derecesine sahip H_2 şemasının tanımlayan uzunluğu, $I(H_2) = 6 - 3 = 3$,
- 8 derecesine sahip H_3 şemasının tanımlayan uzunluğu, $I(H_3) = 10 - 1 = 9$

olur. Benzer şekilde tek sabit pozisyon sahip bir şemanın uzunluğu sıfırdır. 0^{*****} ile ifade edilen bir H şeması tanımlayan uzunluğu, $I(H) = 1-1 = 0$ 'dır.

Şemanın tanımlanan uzunluğu, çaprazlamaya rağmen şemanın yaşama olasılığını hesaplamada faydalıdır.

Bir şemanın ortalama uyumu hesaplanırken, o şemanın örneklerinin ortalama uyumu göz önüne alınır. Yani, bir şemanın ortalama uyumu o şemanın örneklerinin ortalama uyumuna eşittir. n uzunluğuna sahip bir bit dizisi, 2^n farklı şemanın örneğidir. Bu nedenle N diziyeye sahip bir popülasyon 2^n ve $N2^n$ arasında farklı şema içerir. Eğer popülasyondaki tüm diziler özdeş ise 2^n ve tüm diziler farklı ise $N2^n$ şema olacaktır. Bu nedenle, herhangi bir kuşakta GA popülasyondaki N dizinin uyumunu değerlendirirken, gerçekte kapalı olarak daha fazla sayıda şemanın ortalama uyumunu tahmin edebilmektedir. Buna kapalı paralellik adı verilir [52]. Şema, $\{0, 1, *\}$ sembollerinden oluşan bir dizidir. * sembolü, hiçbir zaman GA tarafından işletilmez. Sadece diziler arasındaki benzerlikleri ortaya koyabilmemizi sağlayan gösterimsel bir araç niteliği taşır. Her nesilde, iyiyi belirleyen şemalardaki belirsiz ya da önemsiz elemanlar azalır. Böylece genetik algoritmalar, sonuca, doğru belli kalıplar içinde ilerlerler.

Şemalar ve özellikleri, dizi benzerliklerini ele almada ve sınıflamada önemli gösterimsel araçlardır. Bu özellikler, popülasyondaki şemalarda üreme ve genetik operatörlerin net etkisini incelemede yardımcı olmaktadır. Popülasyondaki şemalar üzerinde, üreme, çaprazlama ve mutasyon ayrı ayrı ve bileşik etkisi aşağıda incelenecektir.

Genetik Operatörlerin Şema Üzerine Etkileri: Popülasyondaki şemalarda üreme ve GA operatörlerinin net etkisi incelenirken, önemli gösterimsel araçlar olan şemalardan ve özelliklerinden faydalanılabilir.

Çoğalmanın Etkisi: Verilen bir t adımından (zamanından); (t+1) adımında yeniden üretim sürecinde H şemasının örneklerinin beklenen sayısı şu şekildedir:

$A(t)$: Popülasyon (Nesil)

$N(H, t)$: t adımında nesilde H şemasına ait örneklerin sayısı,

$f(H, t)$: t adımda H şemasına ait dizilerin uygunluk değerlerinin ortalaması,

$\bar{f}(t) = \frac{\sum_{j=1}^N f_j}{N}$, tüm popülasyonun ortalama uygunluk değeridir.

(t + 1) zamanında (çoğalmada),

$A(t + 1)$ neslinde H şemasının $N(H, t + 1)$ kopyası bulunması beklenir. Buna göre,

$$E(H, t + 1) = N(H, t) \frac{f(H, t)}{\bar{f}(t)} \quad (3.1)$$

olur.

Çaprazlamanın Etkisi

Çaprazlama, diziler arasında yapısal ama aynı zamanda rastlantısal bir bilgi değiş tokuşudur.

$A = 01|1000$ (Birey)

$H_1 = *1*|***0$

$H_2 = ***|10**$ (H_j bireyinin özellikleri)

Bir şemanın çaprazlamayı bozulmadan atlatılması için, kesim noktasının tanımlayan uzunluğun dışında kalması gerekmektedir.

L : Kromozom uzunluğu,

l : Şemanın tanımlayan uzunluğu olmak üzere H şemasına P_c

olasılığı ile t zamanında çaprazlama uygulanırsa, (t + 1) zamanında H şemasının temsil edilme olasılığı,

$$P(H, t + 1) \geq 1 - \frac{P_c I(H)}{l - 1} (1 - P[H, t]) \quad (3.2)$$

olacaktır. Burada, çaprazlanan diğer atanın farklı bir şemanın örneği olma olasılığı $(1 - P[H, t])$ ile ifade edilir.

Mutasyonun Etkisi

Mutasyon, genlerin aleller (genlerin alabileceği sayılar)'inde meydana gelen yapısal bir değişimdir. Bu duruma “*bit olumsuzlama*” adı da verilir. Bir şemanın, mutasyonu bozulmadan atlatılabilmesi için, tüm alellerinin korunması gerekmektedir.

t zamanında o(H) dereceli H şemasına P_m olasılığı ile mutasyon uygulandığında, (t+1) zamanında popülasyonda H şemasının temsil edilme olasılığı,

$$P(H, t + 1) \geq 1 - P_m o(H) \quad (3.3)$$

biçimindedir. Buna göre şema teoremi aşağıdaki gibi ifade edilebilir:

O(H) derecesi ve l(H) tanımlayan uzunluğuna sahip H şemasının, t zamanında ortalama uyumu $f(H, t)$ ve t zamanında tüm popülasyonun ortalama uyumu $\bar{f}(t)$ iken, (t+1) zamanında H şemasını temsil eden örneklerin beklenen sayısı,

$$E(H, t + 1) \geq \left\{ 1 - \frac{P_c I(H)}{l - 1} (1 - P[H, t]) - P_m o(H) \right\} \frac{f(H, t)}{\bar{f}(t)} N(H, t) \quad (3.4)$$

biçimindedir [23]. Çoğu zamanda, şema teoremi;

$$E(H, t+1) \geq \left\{ 1 - \frac{P_c I(H)}{l-1} - P_m o(H) \right\} \frac{f(H, t)}{\bar{f}(t)} N(H, t) \quad (3.5)$$

olarak verilmektedir [54, 65, 66].

Sonuç olarak, kısa tanımlayıcı uzunluklu, düşük dereceli ve yüksek uygunluk (ortalamanın üzerinde uyum değeri) değerine sahip olan şemalar örneklenir, birleştirilir ve potansiyel olarak daha yüksek uygunluk değerine sahip kromozomlar oluşturmak üzere yeniden örneklenir (çoğaltılır).

Şema teoremi, önemli bir sonuç olmasına rağmen, bazı ideal koşullar altında elde edilmiştir. Dizi gösterimi ve genetik operatörler Holland'ın kullandığından farklı olabilir. Yapı parçaları hipotezi birçok durumda güvenilir bulunmasına rağmen, bu hipotezde dizi gösterimlerine ve genetik operatörlere dayanmaktadır. Bu hipotezin sağlanmadığı problemleri bulmak zor değildir. Bu tür problemlere aldatıcı problemler adı verilir. Buna rağmen bu teori, basit genetik algoritmaların işleyişi için temel dayanaktır [62].

3.5. Genetik Algoritmaların Direk Arama Yöntemlerinden Farkları

GA, arama problemlerinde oldukça etkilidir, çözüm kümesinden optimum veya optimuma yakın çözümleri bulabilir. GA'lar adım adım genetik operatörleri uygulayarak, uygun toplulukta yeni nesiller üreterek en iyi çözümlere ulaşılmasını sağlar. GA'lar geçmiş yıllarda yapılan bazı çalışmalarda özellikle mühendisler tarafından sıklıkla kullanılan algoritmalar haline gelmiştir. Holland'ın Machine Learning (Makine Öğrenme) sistemlerinde yardımcı olarak kullanılmış, daha sonra GA'nın çok sayıda kollara ayrılmış gaz borularında gaz akışını düzenlemek ve kontrol etmek için uygulanması tamamlanmıştır (Goldberg, 1989).

Genetik algoritmaların direk arama tekniklerine göre bazı farklılıkları vardır. Bunlar;

- a) Genetik algoritmaların işleyişi modeldeki parametrelerin kodlanmasıyla başlar,
- b) Genetik algoritmalar çözüm kümesine bir tek noktadan değil, birçok noktadan başlar.
- c) Genetik Algoritma, deterministik yöntemlerdeki gibi varsayım kümesine ihtiyaç duymaz. Algoritmanın kendine özgü işleyişi modelden modele farklılık göstermez. Modeldeki parametreler uygun bir şekilde tanımlandığında, GA kendi terminolojisi ile çözüme ulaşır.
- d) GA'lar sebep – sonuç ilişkisi içinde rastgele çalışan operatörler kullanır. Ancak burada, sebep – sonuç ilişkisi kurallarını değil olasılık kurallarını kullanır.

3.6. Genetik Algoritma ve Uygulama Alanları

GA'lar uygulamadaki kolaylıkları nedeniyle birçok alanda yaygınlıkla kullanılan bir metottur. GA'lar geleneksel optimizasyon yöntemlerine göre zor süreksiz ve gürültü içeren fonksiyonları çözümede daha etkindirler. GA'lar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir. Amaç fonksiyonunun süreksiz olması durumlarda, süreksiz noktalarda türev alma işlemi yapılamayacağından türeve dayalı optimizasyon yöntemleri uygulanamamaktadır. GA'lar türev ve diğer yardımcı bilgilere ihtiyaç duymadığından bu tür problemlerde geleneksel yöntemlere göre önemli bir üstünlüğe sahiptir.

Genetik algoritmaların başlıca uygulama alanları optimizasyon, otomatik programlama ve bilgi sistemleri, mekanik öğrenme, ekonomik ve sosyal sistem modelleri başlıkları altında toplanabilir.

Sınırlı kaynakların etkin kullanımı problemleri genel olarak işgücü, tedarik ve bütçe ile ilgilidir. Bu tip problemlere örnek olarak, gezgin satıcı problemi, rota problemi, kargo yükleme problemi, atama problemi, atölye çizelgeleme problemi, yerleşim tasarımı problemi örnek verilebilir. Bu tip problemlerde, incelenen değişken sayısının artması çözüme ulaşma zamanını da artırmaktadır. Çözüm uzayının tamamının taranmasını gerektiren geleneksel optimizasyon yöntemlerinde problem çözümü değişken sayısının artmasıyla imkansız hale gelebilmektedir. GA ise çözüm uzayının yalnızca belli bir kısmını taradığı ve eş zamanlı arama yaptığı için bu tip problemlerde çözüme daha kısa sürede ulaşabilmektedir [64].

4. TERMOELEKTRİK MODÜLLERİN PARAMETRELERİNİN GENETİK ALGORİTMA İLE OPTİMİZASİYONU VE MAKSİMUM SOĞUTMA KAPASİTESİNİN HESAPLANMASI

Bir termoelektrik cihazında bir termoelektrik modül kullanıldığında, modülün belirli bir soğutma kapasitesine sahip olanları seçilir. Buna göre uygun modüllerin dizayn edilmesi gerekir. Modül maliyetinin yüksek olması ve tamir edilememesi optimum parametrelere sahip modüllerin doğru seçilmesi önem arz etmektedir. Bu yüzden üretime geçmeden önce termoelektrik modülün optimizasyonu yapılarak tasarlanması modül seçim probleminin çözümüne büyük katkı sağlamaktadır [12].

4.1. Problemin Genel Tanımı

Termoelektrik modülün performansı, modüle uygulanan akıma, sıcak yüzeyinin sıcaklığına ve geometrik yapısına bağlıdır. Bu parametreler, modülün hacmine, termoelementlerin boyutlarına ve yarıiletkenlerin parametrelerine göre değişmektedirler. Bundan dolayı bir modülün optimizasyonu iki grup şeklinde yapılmalıdır. Birinci grup optimizasyon yarıiletkenin parametrelerinin optimizasyonunu içerir. İkinci grup optimizasyon ise termoelektrik modül kolların boyutlarının optimizasyonu içermektedir.

Optimizasyon: Optimizasyon temel bir tanımlama ile verilen kısıtlar altında en iyi çözümün bulunması işidir. Optimizasyon problemlerinin çözümü belirli sınırlamaları sağlayacak şekilde matematiksel ifadelerle veya kurallara dayanan algoritmalarla mümkün olmaktadır [43]. Optimizasyon problemlerinin çözümünde, farklı teknikler kullanılmaktadır. Bu tekniklerden birisi yapay zekânın bir alt tekniklerinden genetik algoritmadır. Genetik algoritmalar reel hayata ilişkin problemlerin çözümünde geleneksel optimizasyon problemlerine oranla daha başarılı sonuçlar üretirler.

Bu çalışmada bir modülün parametrelerinin optimizasyonu için genetik algoritma yöntemi kullanılarak gerçekleştirilmiştir [19].

4.2. Matematiksel Modelleme

Soğutma kapasitesinin değerini bulmak için aşağıdaki matematiksel eşitlikler kullanılmıştır [10,13].

Soğuk yüzeyin soğutma kapasitesi ($Q_C (W)$);

$$Q_C = 2N \left[\alpha IT_C - ((I^2 \rho) / (2(A/H))) - k\Delta T(A/H) \right]$$

Sıcak yüzeyin ısıtma kapasitesi($Q_H(W)$);

$$Q_H = 2N \left[\alpha IT_H + ((I^2 \rho) / (2(A/H))) - k\Delta T(A/H) \right]$$

Performans katsayısı (COP),

(4.1)

$$COP = \frac{Q_C}{Q_H - Q_C} = \frac{Q_C}{P}$$

TE modülün tükettiği güç ($P,(W)$),

$$P = 2N \left(\frac{I^2 \rho}{G} + \alpha I \Delta T \right)$$

Burada

A : Alan, (mm²)

H : Kol yüksekliği, (mm)

İken çizelge 2.1'deki parametrelere ilave olarak

S :Kollar arasındaki boşluk alanı, (mm²)

parametreleri eklenmiştir.

$$\text{Maliyet} = d(N \times A \times L) \times (\text{kg başına maliyet}) \quad (4.2)$$

Ayrıca d, termoelektrik maddenin yoğunluğudur.

Hacim(V)=Kolların hacmi + Kollar arasındaki boşlukların hacmi;

$$V = 2.N.A.H + (2.N-1).\delta.H \quad (4.3)$$

den " δ " çekildiğinde

$$\delta = \frac{V-2.N.A.H}{(2N-1).H} \quad (\text{mm}^2) \text{ olur.}$$

4.3. Genetik Algoritmanın Uygulanması

Problemin tanımlanması ve matematiksel modelleme sonrasında parametrelerin genetik algorithmada nasıl tanımlanacağı bu bölümde açıklanmaktadır.

4.3.1. Kromozomların oluşturulması

Problemin çözümünde var olan parametreler, termoelektrik soğutucunun kol boyutları (kol alanı, kol uzunluğu, kol sayısı), akım, sıcak yüzeyin sıcaklığı, sıcaklık farkı, kollar arasındaki boş alan, Hacim olmak üzere optimizasyon değişkeni olarak ele alınmıştır. GA'da parametreler kromozomların genleri olarak tanımlanmaktadır. Burada Popülasyon birçok sayıda kromozomdan oluşur. Bir kromozomun (bireyin) GA'da nasıl tanımlandığı büyük önem taşır. Bu tanımlama çözüm uzayının sınırlarını ve olasılık sayısını etkiler.

Uygulamada parametreleri tamsayı şeklinde kodlanması uygun görülmüştür.

Çizelge 4.1. TEGAO 5'in optimizasyon amaçlı seçtiği 300 adet TE modülün parametrelerinden oluşan örnek listesi

TE modüller	I_{\max} (amper)	A (mm^2)	H (mm)	Hacim (mm^3)	N	T_H (K)	T_C (K)	ΔT (K)	(mm^2)
1.TE. modül	6,04	3,24	3,8	2164	176	300	300	0	0,1
2.TE. modül	1,66	0,62	0,3	23	122	303,06	301,93	1,13	1,1
3.TE. modül	6	4,5	3,8	187,72	11	310,85	307,12	3,73	0,2
.
.
.
300.TE modül

N, popülasyon büyüklüğünü göstermektedir. Veri olarak 300 termoelektrik modül alındığında popülasyon büyüklüğü 300 olmaktadır. Bir kromozom içerisindeki her bir gen, bir termoelektrik modülün yapısını oluşturan parametreleri içermektedir. Kromozom üzerinde genlerde temsil edilen parametreler;

- I : Akım, (amper)
- A : Alan, (mm^2)
- H : Kol yüksekliği, (mm)
- T_H : Sıcak yüzeyin sıcaklığı, (K)
- : Sıcaklık farklılığı, (K)
- : kollar arasındaki alan, (mm^2)
- V : Hacim, (mm^3)

4.3.2. Uygunluk Fonksiyonunun Hesaplanması

Genetik algoritma, evrim teorisinin temelini oluşturan “En dayanıklı, çevreye en çok uyum sağlayan birey yaşar” prensibine göre işlemektedir. Ortama uyum sağlayan bireyler yaşar ve bu bireylerden yeni bireyler oluşur. Popülasyon devamlı olarak iyi özellikteki genleri bulundurmaya çalışır. GA'da

farklı bireylerdeki iyi özelliklerin, çaprazlama sonucu oluşacak yeni bireylere aktarılacağı böylece en iyi çözüme doğru ilerleneceği temel alınmıştır. Çaprazlama sonucunda iyi bireylerden kötü bireyler de elde edilebilir. Kötü bireyler uygunluklarına göre değerlendirilip popülasyondan atılır ve nesil devam eder. GA'nın popülasyondaki hangi bireyin ortama uygun olduğu, iyi özellikler taşıdığını anlayabilmesi için matematiksel bir fonksiyona ihtiyacı vardır. Bir bireyin aranan çözüme ne kadar uygun ya da yakın olduğunu o bireyin "uygunluk değeri" belirler [19, 20,61, 62]

Bu çalışmada Sınırlanmış hacimlerde termoelektrik modülün QC, COP, P ve maliyet elde etmek için Eş. 4.1, Eş. 4.2 ve Eş. 4.3'den uygunluk değerleri (amaç fonksiyon) hesaplanmıştır. uygunluk değeri maksimum soğutma kapasitesi, performans katsayısı, güç ve maliyet olarak tanımlanmıştır.

4.3.3. Seçme operatörü

Kodlamaya karar verilip 300 termoelektrik modül (başlangıç popülasyonu) oluşturulduktan sonra yeni popülasyona geçebilmek için seçimin nasıl yapılacağı belirlenmelidir. İstenilen uyuma sahip bireyler daha büyük bir olasılıkla seçilmelidir. Genetik algoritmada çeşitli seçim planları önerilmiştir. Maksimum soğutma kapasitesi ve optimal boyutları bulmak için yapılan genetik algoritma uygulamasında seçim yöntemlerinden Elitizm yöntemi uygulanmıştır. En uygun bireyleri kopyalayarak bir sonraki nesile aktarılması işlemi yapar. İstenilen sayıda kromozomun bir sonraki nesile kopyalanması işlemi yapan basit bir işlemdir. Buna rağmen popülasyon büyüklüğünün korunmasını sağlayan önemli bir operatördür. Bu çalışmada elde edilen 300 termoelektrik modül içerisinde en yüksek soğutma kapasitelerine sahip olan 16 termoelektrik modülü (16 bireyi) kopyalayarak bir sonraki nesile aktarıma işlemi yapılmıştır. Böylece en iyi soğutma kapasitelerine sahip olan termoelektrik modüllerin seçilmesi sağlanmıştır. Bu yöntemin avantajı, en yüksek soğutma kapasitesi değerine sahip olan termoelektrik modüllerin

örnekleme hatasını ya da genetik operatörlerin kullanılması ile kaybolmasını önlemektir.

4.3.4. Çaprazlama

Çaprazlama operatörünün uygulanması ile yeni nesile iyi özelliklerin taşınması amaçlanmaktadır. Çaprazlamada önemli olan P_c çaprazlama olasılığıdır. Popülasyon büyüklüğüne bağlı olarak De Jong'un kabullerine göre popülasyon genişliğine göre çaprazlama olasılığı 0,60 kabul edilmiştir. Çalışmada popülasyon genişliği 300 olarak kabul edildiğinden, çaprazlama oranı 0,60 olarak hesaplanmaktadır. Çaprazlama genetik algoritmaya dayalı çalışmalarda en önemli operatördür. Çaprazlama ile kromozomlardaki iyi özellikler birleştirilerek daha iyi çözümler yaratması beklenir. Çaprazlama operatörünün uygulanması ile yeni nesile iyi sonuçların taşınması amaçlanmaktadır. Uygulamada tek noktalı çaprazlama tercih edilmiştir. Çaprazlama işlemi rastgele iki termoelektrik modülün seçimiyle başlanmıştır. Rastgele seçilen iki termoelektrik modülün, rastgele seçilen iki parametre değerleri yer değiştirilmiştir. Bu işlem, 300 termoelektrik modül oluşuncaya kadar tekrar edilmektedir. Her çaprazlamadan sonra elde edilen sonuçlar yeni diziye aktarılmaktadır. Şekil 4.1. iki TE modülün parametreleri arasında tek nokta çaprazlamayı göstermektedir. Burada 6. gen çaprazlamaya uğruyor.

Tek nokta çaprazlama

I.Ata

I	A	H	N	ΔT	T_H	δ
0,89	3,76	0,42	14	3,40	443,78	1,12

II.Ata

I	A	H	N	ΔT	T_H	δ
6,58	4,58	0,49	31	2,58	404,14	0,79

I.Yavru

I	A	H	N	ΔT	T_H	δ
0,89	3,76	0,42	14	3,40	404,14	1,12

II.Yavru

I	A	H	N	ΔT	T_H	δ
6,58	4,58	0,49	31	2,58	443,78	0,79

Şekil 4.1. İki TE modülün parametreleri arasında tek nokta çaprazlama

4.3.5. Mutasyon

Mutasyon kromozomdaki bir yada daha fazla genin değerlerinde değişiklik yapılmasıdır. Genetik alıtmada her çözümün her geninin mutasyondan etkilenme olasılığı vardır. Buna da mutasyon olasılığı denir mutasyon olasılığı olan P_m , 0 ile 1 arasında ($0 < P_m < 1$) belirlenmiştir. 0 ile 1 arasında rastgele bir sayı seçiliyor, eğer seçilen sayı 1/100'e eşit veya 1/100'den küçük ise modül mutasyona uğrar, aksi takdirde uğramamaktadır.

4.3.6. Rulet tekerliđi

Son iřlem olarak 300 modülün sođutma kapasitelerinin deđerleri toplanmaktadır ($T = \sum_{i=1}^{300} f_i$). Sonra 0 ile T arasında rastgele iki sayı çekiliyor.

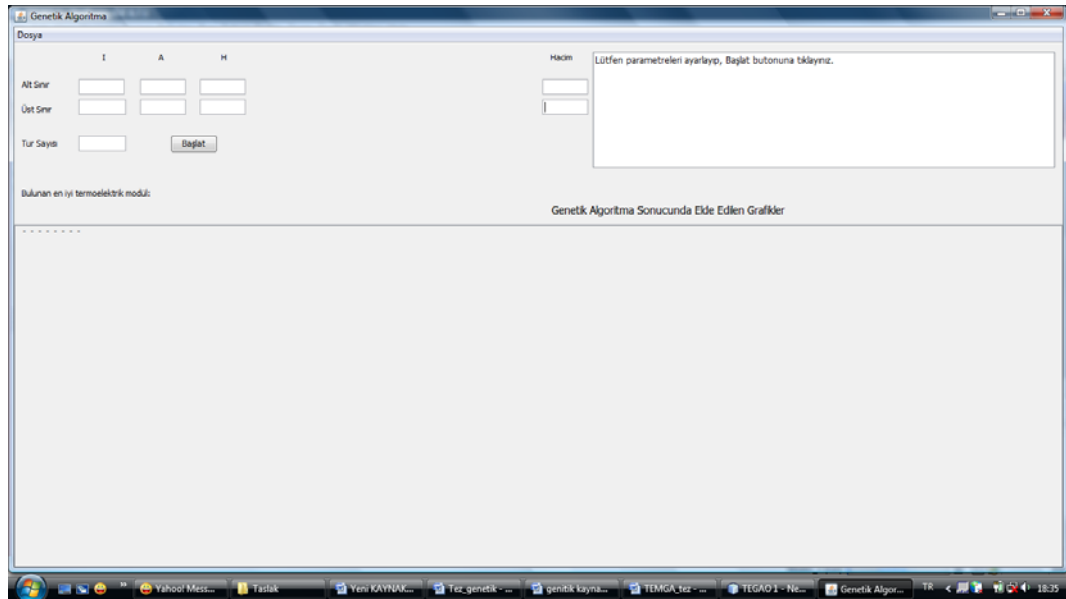
Bu sayıya denk gelen modüller aprazlama iřlemine tabi tutuluyor. Bylece birinci iterasyon tamamlanmıř oluyor

4.3.7. Durdurma kriteri

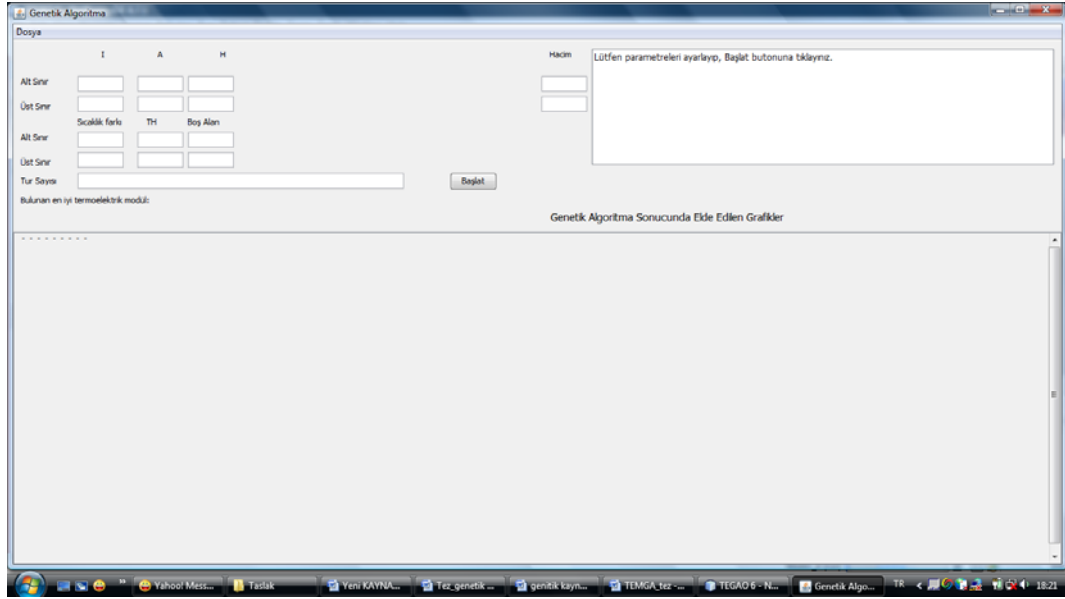
Genetik algoritma iteratif bir sre olduđu iin bu tekrarlanmanın ne zaman biteceđi jenerasyon sayısı ile belirlenmelidir. Program iřlediđinde, ne kadar yavrunun oluřacađı kontrol parametresi olarak belirlenmesi gerekmektedir. Bu uygulamada jenerasyon sayısı 50-100'e kadar iteratif iřletilebilir olduđu gzlemlenmiřtir. İterasyon sayısının belirlenmesi ve izlenmesi ilerleyen blmlerde test edilmiřtir.

5. GELİŞTİRİLEN TERMOELEKTRİK MODÜLÜN GENETİK ALGORİTMAYLA OPTİMİZASYONU SAĞLAYAN (TEGAO) PROGRAMLARIN YAPISI

Genetik algoritma kullanılarak dördüncü bölümde verilen matematiksel eşitliklere göre, programlama dili olan Java ile NetBeans 7.6.1. ortamında TEGAO 1, TEGAO 2, TEGAO 3, TEGAO 4, TEGAO 5 ve TEGAO 6 programları geliştirilmiştir. TEGAO programlarında başlangıç popülasyonu, rastgele seçilen 300 termoelektrik modül parametreleri ile belirlenmiştir. Bu bölümde programlar hakkında genel bilgi verilmekte ve menüler tanıtılmaktadır. Ayrıca bu bölümde örneklerle programlar çalıştırılmış ve elde edilen sonuçlar değerlendirilmiştir. Şekil 5.1 ve Şekil 5.2’de yazılan TEGAO programlarının menü görüntüsü verilmektedir.



Şekil 5.1. TEGAO 1, 2 ve 3 program menüsünün görüntüsü



Şekil 5.2. TEGAO 4, 5 ve 6 program menüsünün görüntüsü

Veri butonlarının işleyişi

Sınırlanan hacimde, optimizasyon yapılması istenilen parametrelerin aralık değerleri Şekil 5.1 ve Şekil 5.2'deki alanlara yazılmaktadır. Burada kullanıcı tarafından sınırlanan aralık, alt sınır ve üst sınır olarak girilmekte, ondalık sayı ile de belirlenebilmektedir.

Tur sayısı: İterasyon sayısını belirtir. 50 veya 100 tur olarak ayarlanmıştır.

Başlat butonu: Programın çalıştırıldığı butondur. Aralık değerleri yazılıp "Başlat" butonuna basılır.

Ekranın sağ üst penceresi: Ekranın sağ üst tarafındaki pencerede "Lütfen parametreleri ayarlayıp, Başlat butonuna tıklayınız." yazmaktadır. Parametre değerleri yazıldıktan sonra "Başlat" butonuna basıldığında ekranın sağ tarafında 300 Başlangıç popülasyonu oluşur. Oluşan 300 termoelektrik modül üzerinde genetik algoritmanın özellikleri ve genetik operatörlerden seçme, çaprazlama ve mutasyon uygulanır. Bu işlem 50 kez tekrarlanarak toplam

60000 termoelektrik modül yazılır. Bu işlemler birkaç saniye içinde gerçekleşmektedir.

Ekranın alt penceresi: “Bulunan En İyi Termoelektrik Modül” yazısının altında, programın çalıştırılması sonucunda bulunan maksimum soğutma kapasiteli termoelektrik modülün parametrelerinin optimal değerleri ve performans katsayı yazılmaktadır.

Programın çalışması sonucunda karşılaşılan şekiller: Menüdeki şekiller genetik arama sürecini göstermektedir. Yani genetik algoritmanın işlemini grafiksel olarak göstermektedir. Şekillerde görüldüğü gibi 300 termoelektrik modülde 50 kez genetik operatörler yapılmıştır. Yaklaşık 20. iterasyondan sonra maksimum soğutma kapasitesinin ve optimal değerlerin hiç değişmediği görülmektedir. Bu sonuçtan, optimal değerlerin bulunduğu ve maksimum akım ve maksimum soğutma kapasitesi ve optimal değerler elde edildiği anlaşılmaktadır.

5.1. TEGAO 1 Programının Menüsü ve Çalışması

TEGAO 1 program menüsü, sol tarafında 3 parametre ve her parametrenin alt ve üst sınırları, hacim, “tur sayısı” ve “başlat” butonu olarak toplam 10 butondan oluşmaktadır. Menüde sağ tarafında, kullanıcının belirlediği hassasiyet aralığına göre program çalıştığında genetik algoritma yöntemiyle termoelektrik modülleri oluşmaktadır. Oluşan termoelektrik modüllerin parametre değerleri, soğutma kapasiteleri ve COP değerleri yazılmaktadır.

5.1.1. TEGAO 1 ile maksimum soğutma kapasitesinin hesaplanması

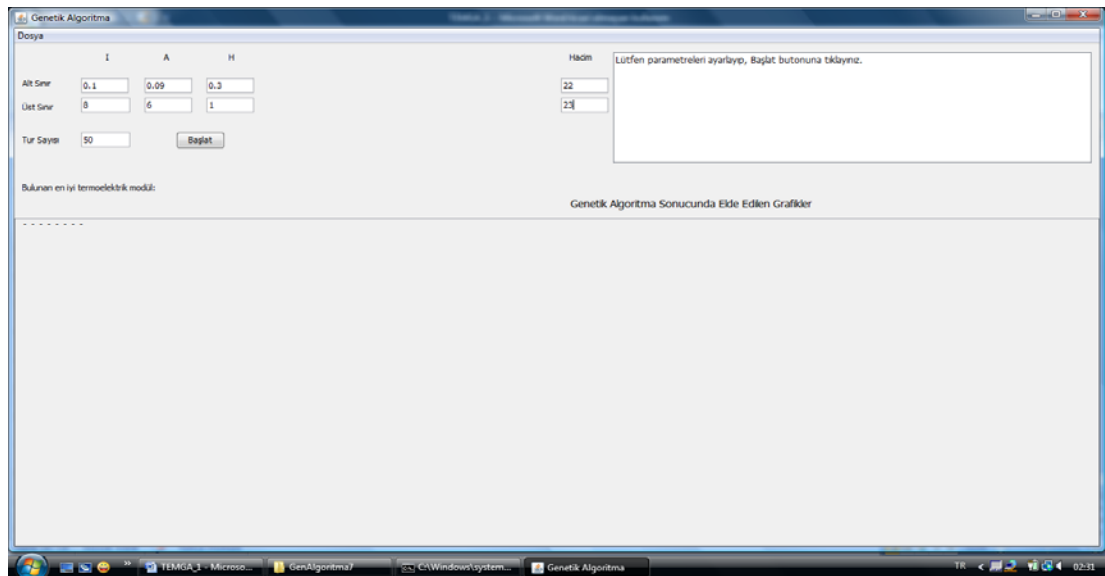
TEGAO 1 programı ile modülün sınırlanan hacminde, $\Delta T=0$ ve $T_H= 300K$ kabul edilerek soğutma kapasitesinin ve COP'un maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği ve kollardan geçen akıma göre optimizasyon işlemi yapılmaktadır. Programın çalıştırılmasındaki ilk adım, kullanıcının istediği termoelektrik modülün hacminin belirlenmesidir. Burada

örnek olarak hacim $22 - 23 \text{ mm}^3$ arasında bir termoelektrik modülü ele alınmıştır. Genetik algoritmalar yöntemiyle maksimum soğutma kapasitesi ve geometrik optimizasyonun hesaplanması için termoelektrik modülün kol boyutları ve kollardan geçen akım sınırlanmıştır. Çizelge 5.1’de seçilen bir TE modülün parametrelerinin aralıkları yazılmaktadır. Akımın alt sınır 0,1 amper ve üst sınır 8 amper, kol alanının alt sınır $0,09 \text{ mm}^2$ ve üst sınır 6 mm^2 , kol uzunluğunun alt sınır 0,3 mm ve üst sınır 1 mm olarak belirlenmiştir.

Çizelge 5.1. TEGAO 1 programında seçilen örnek uygulama kriterleri

TE.modülün Hacmi(mm^3)	Akım I(A)	Alan A(mm^2)	Yükseklik H(mm)
22 - 23	0,1 - 8	0,09 - 6	0,3 - 1

Çizelge 5.1’de belirlenen değerlerin program menüsüne girilmiş gösterimi Şekil 5.3’de verilmektedir.



Şekil 5.3. Parametre değerleri girilmiş TEGAO 1 programın menüsü

Burada sınırlanmış bir hacimdeki termoelektrik modülü optimize etmek için iki farklı parametre, sabit parametreler ve değişken parametreler olarak sınıflandırılmıştır. Burada T_H ve ΔT sabit parametre olarak tanımlanmaktadır.

Kol alanı, kol yüksekliği ve akım deęişken parametreleri olarak ele alınmıştır. Optimizasyon işlemi için bu 3 parametrenin deęerleri belirlenen sınırlar aralığından rastgele seçilmiştir. Seçilen bu deęerler Eş. 4.1 ve Eş. 5.1'de yerlerine konularak soęutma kapasitesinin (Q_{CG}) deęeri hesaplanmıştır. Çizelge 5.2 rastgele seçilen 300 termoelektrik modülden ilk 30 modülün parametre deęerleri ve soęutma kapasiteleri göstermektedir.

$\Delta T = 0$ ve $T_H=300$ olmak üzere Seebeck katsayısı (α) ve elektriksel özdirenç (ρ) aşığıdaki gibi genelleştirilmektedir [23].

$$\alpha = 2.02 \cdot 10^{-4} \text{ (V/K) , } \quad \rho = 1.01 \cdot 10^{-2} \text{ (}\Omega \text{ mm)} \quad (5.1)$$

Çizelge 5.2. Rastgele seçilen 300 termoelektrik modülden ilk 30 modülün parametre değerleri ve soğutma kapasiteleri

Akım (I,A)	Alan (A,mm ²)	Yükseklik (H,mm)	Termoelement Sayısı (N)	Soğutma kapasite (Q _{CG} ,W)	COP	Hacim (V,mm ³)
7,55	3,88	0,49	12	11,32	0,84	22,91
7,52	4,70	0,95	5	4,69	0,84	22,44
1,85	4,42	0,36	14	4,76	6,16	22,32
6,00	3,39	0,60	11	7,42	0,81	22,08
1,78	1,84	0,32	39	11,49	2,45	22,69
4,94	1,85	0,34	36	18,62	0,56	22,69
0,71	3,76	0,33	18	2,45	14,52	22,36
2,84	1,57	0,44	33	12,34	0,98	22,81
5,50	3,14	1,00	7	4,48	0,71	21,92
2,90	5,14	0,92	4	2,01	3,36	18,88
3,26	3,55	0,87	7	3,60	1,92	21,55
6,81	2,64	0,39	22	15,79	0,56	22,96
1,49	1,20	0,75	25	5,43	1,38	22,46
4,80	1,26	0,33	55	17,31	0,25	22,60
0,78	3,48	0,94	7	1,03	9,05	22,99
3,67	5,46	0,50	8	5,08	3,38	21,82
1,82	2,3	0,76	13	3,90	2,43	22,65
4,77	1,54	0,77	19	5,93	0,25	22,56
3,43	0,91	0,34	73	16,38	0,25	22,70
7,08	2,59	0,72	12	7,13	0,36	22,38
1,49	4,44	0,37	13	4,63	5,82	21,47
2,47	2,59	0,92	9	3,45	1,79	21,41
0,53	5,62	0,65	6	0,62	25,46	21,75
0,87	4,38	0,50	10	1,67	12,55	22,04
2,21	5,86	0,37	10	4,11	6,86	21,44
1,25	5,67	0,95	4	0,94	9,26	21,61
2,90	2,91	0,40	19	6,86	3,31	21,93
4,89	3,14	0,46	16	11,00	1,21	22,87
0,61	5,74	0,97	4	0,47	19,63	22,18
4,23	4,91	0,84	5	3,40	2,10	20,65

Genetik algoritma işleyiş sürecine göre seçim yöntemi uygulanır. Burada seçim yöntemlerinden elitist seçim yöntemi uygulanmıştır. Bu aşamadan sonra genetik algoritmanın operatörlerinden, çaprazlama ve mutasyon operatörleri uygulanmıştır. Çalışmada tek noktalı çaprazlama tercih edilmiştir.

Çaprazlama işleminden sonra mutasyon işlemi başlamıştır. Burada mutasyon olasılığı olan P_m , 0 ile 1 arasında ($0 < P_m < 1$) belirlenmiştir.

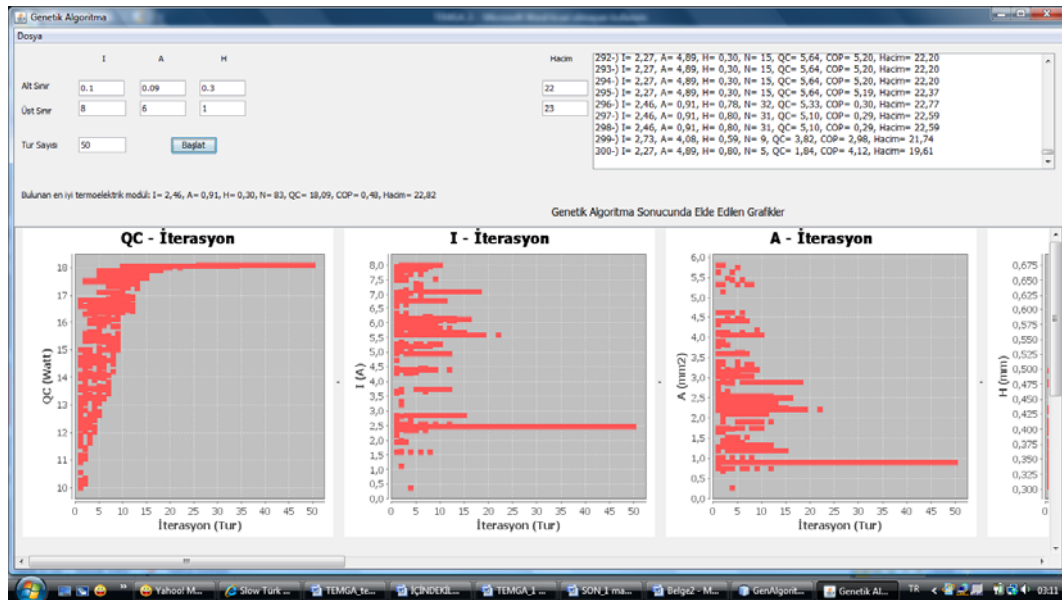
Bu kriterlere göre program çalıştırıldığında 50 iterasyondan sonra menünün sağ üst penceresinde 60000 termoelektrik modül oluşmaktadır. Çizelge 5.3, 50 defa genetik operatör işlemlerine uğradıktan sonraki birkaç termoelektrik modülün parametre değerlerini ve soğutma kapasitelerini göstermektedir.

Çizelge 5.3. 50 defa genetik operatör işlemlerinden sonra birkaç termoelektrik modülün parametre değerlerini ve soğutma kapasiteleri

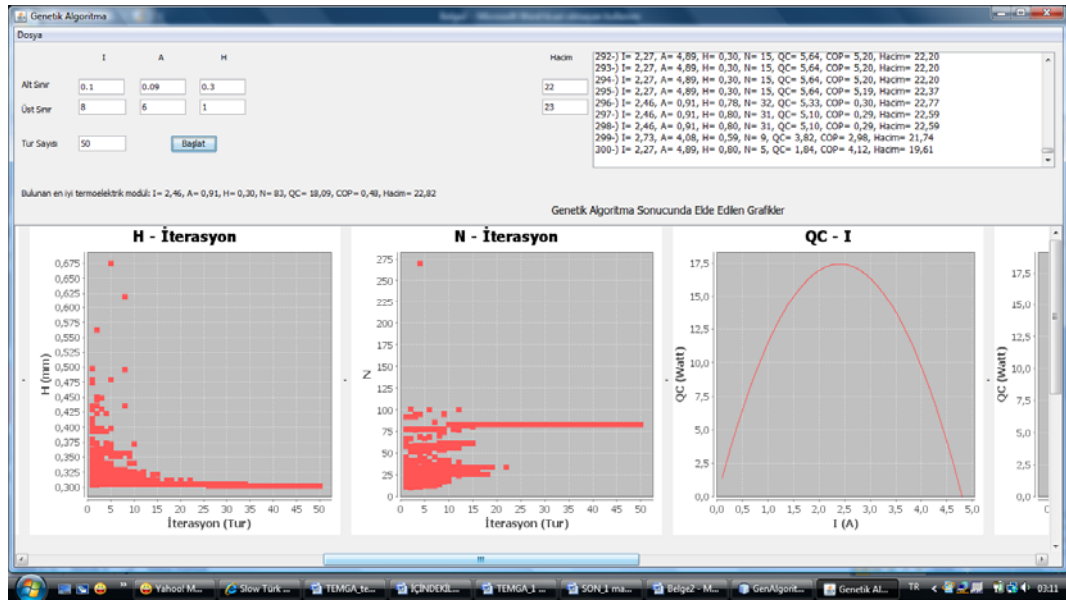
Akım (,IA)	Alan (A,mm ²)	Yükseklik (H,mm)	Termoelement Sayısı (N)	Soğutma kapasite (Q _{CG} ,W)	COP	Hacim (V,mm ³)
2,46	0,91	0,30	83	18,09	0,48	22,82
2,46	0,91	0,30	83	18,07	0,48	22,99
6,10	2,50	0,30	30	17,91	6,58	22,71
6,10	2,50	0,60	30	17,89	0,81	22,08
2,46	1,17	0,30	65	17,43	0,75	22,91
2,46	0,91	0,59	83	15,51	0,36	22,69
2,46	0,91	0,61	83	15,33	0,35	46,23
2,46	0,91	0,78	83	13,81	0,30	59,07
2,46	0,91	0,80	83	13,65	0,29	60,49
2,46	0,91	0,87	83	13,07	0,27	65,34
6,10	1,47	0,30	51	12,07	0,14	22,66
2,46	0,91	0,47	54	10,81	0,41	22,99
2,46	2,96	0,30	25	9,40	2,69	22,41
2,46	0,91	0,30	42	9,15	0,48	11,55
2,46	0,91	0,59	42	7,85	0,36	22,61

Burada 50 iterasyondan sonra bulunan maksimum soğutma kapasitesi ve optimal parametre değerleri Çizelge 5.3'ün ilk satırında yazılmaktadır. Dolayısıyla 23 mm³ hacimdeki bir TE. modülün maksimum soğutma kapasitesi 18,09 W, maksimum akım 2,46 amper, optimal alan 0,91mm², optimal kol yüksekliği 0,3 mm, optimal kol sayısı 83 ve performans katsayısı 0,48 olarak bulunmuştur.

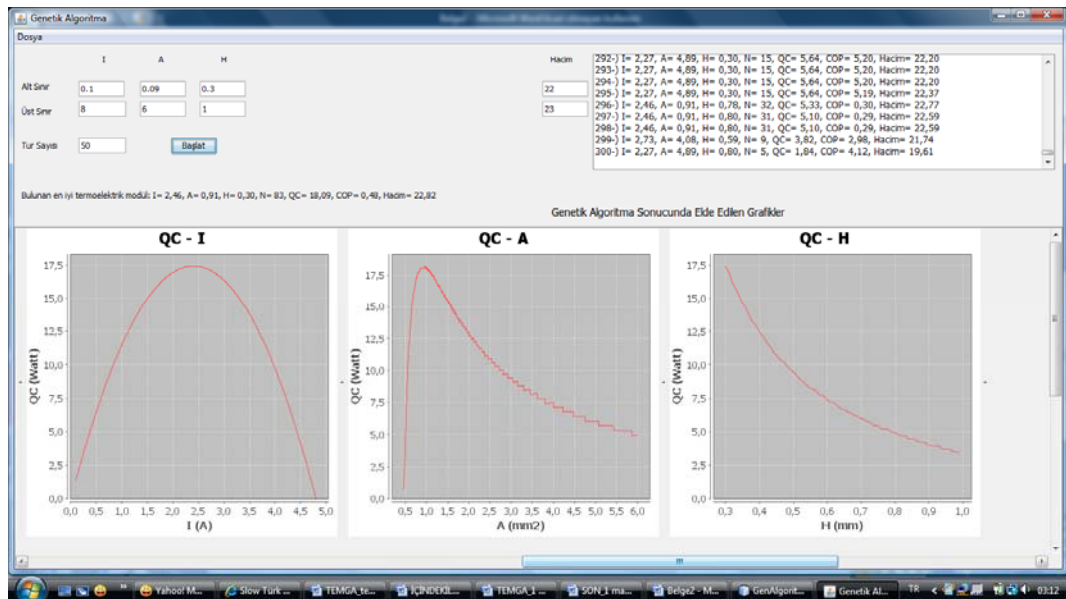
Menüdeki şekiller: Menüdeki şekiller genetik arama sürecini göstermektedir. Şekillerde görüldüğü gibi başlangıç popülasyonu olarak 300 termoelektrik modülün 50 kez genetik operatörlerin yapıldığını göstermektedir. Yaklaşık 20. iterasyondan sonra maksimum soğutma kapasitesinin bulunduğunu ve kol alanının, kol yüksekliğinin ve kol sayısının artık hiç değişmediği görülmüştür. Demek ki kol alanının, kol yüksekliğinin ve kol sayısının optimal değerleri bulunmuştur ve maksimum akım ve maksimum soğutma kapasitesi elde edilmiştir. Şekil 5.4'de TEGAO 1 programında tercih edilen aralıklar için maksimum soğutma kapasitesi, maksimum akım ve optimal kol alanına ilişkin sonuç ekran görüntüsü verilmektedir. Şekil 5.5'de ise TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre ekran görüntüsü verilmektedir. Şekil 5.6'da TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre ekran görüntüsü verilmektedir.



Şekil 5.4. TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, maksimum soğutma kapasitesi, maksimum akım ve optimal kol alanının görüntüsü



Şekil 5.5. TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü

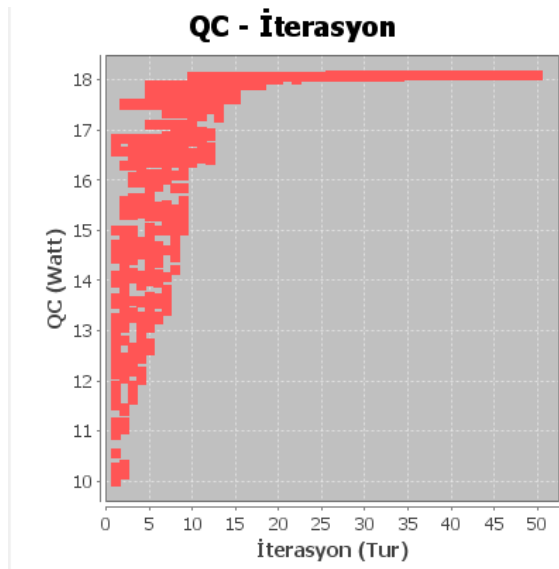


Şekil 5.6. TEGAO 1 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü

Genetik algoritmaların işlemi grafiksel olarak gösterildiğinde, aşağıdaki şekillerde genetik arama sürecini gösterebilmektedir. Şekillerde görüldüğü gibi 300 termoelektrik modül 50 kez genetik operatörlerin yapılmasını

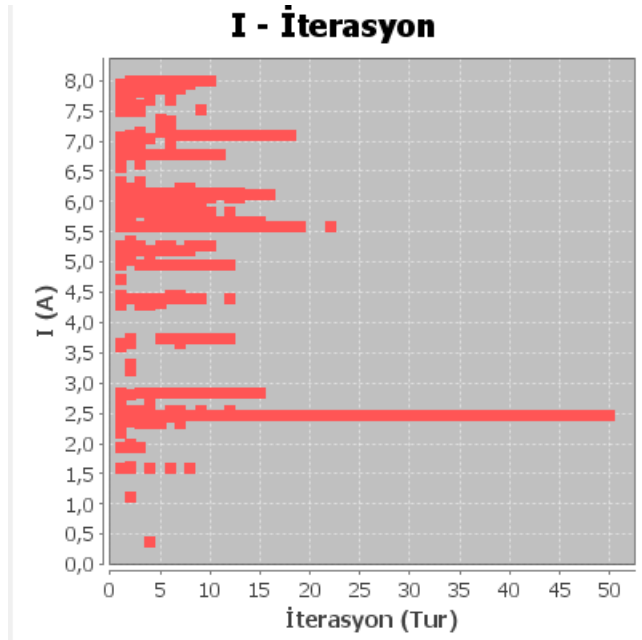
istediğimizde yaklaşık 20. İterasyondan sonra maksimum soğutma kapasitesi, kol alanının, kol yüksekliğinin ve kol sayısının artık hiç değişmediğini görülmektedir. Demek ki kol alanının, kol yüksekliğinin ve kol sayısının optimal değerleri bulunmuştur ve maksimum akım ve maksimum soğutma kapasitesi elde edilmiştir.

Şekil 5.7'de, sınırlanmış bir hacimde termoelektrik soğutucunun genetik algoritma yöntemiyle optimizasyon sonucu maksimum soğutma kapasitesi 18,09 W olarak bulunduğu gösteriştir. Burada optimal boyutlar; kol uzunluğu 0,3 mm, kol alanı 0,91 mm², kol sayısı 83 ve akım 2,46 amper olmuştur. Genetik arama işlemi birkaç dakikada tamamlanmıştır. Analitik sonuç ile genetik aramadan elde edilen bir maksimum soğutma kapasitesinin karşılaştırması, genetik sonucun analitik sonuca tam olarak benzediğini göstermektedir.



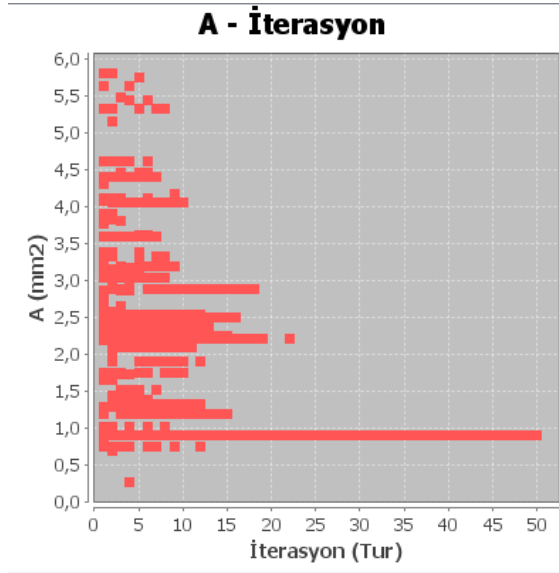
Şekil 5.7. TEGAO 1 program ile Q_C 'nin 50 iterasyona(turuna) göre yaklaşımı

Şekil 5.8, genetik arama süresince maksimum akımın 2,46 ampere yakınlığına göstermektedir. Bu arama, uygun çözümlerin rastgele bir popülasyonu ile başlamakta ve bu popülasyon yaklaşık 20 iterasyon sonra bir noktada birleşmektedir.



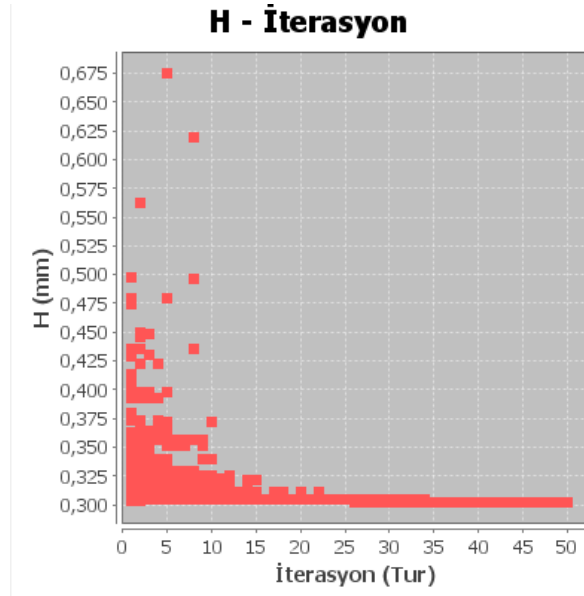
Şekil 5.8. TEGAO 1 program ile akımın 50 iterasyon (turuna) göre yaklaşımı

Şekil 5.9 genetik arama süresince kol alanın 0,91'de yakınlaştığını göstermektedir.



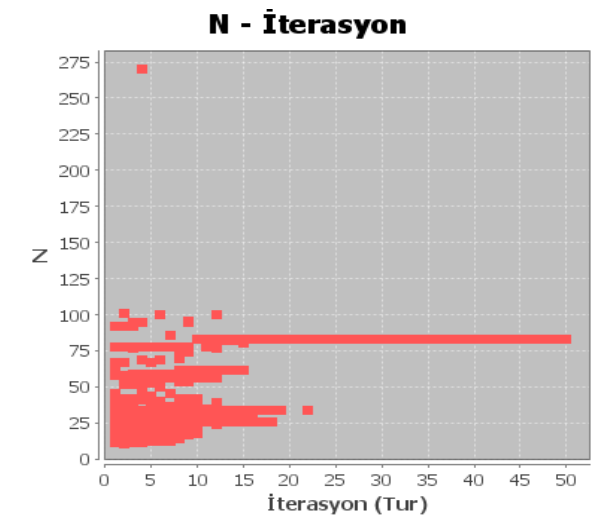
Şekil 5.9. TEGAO 1 program ile kol alanının 50 iterasyona (turuna) göre yaklaşımı

Şekil 5.10'da verilen aralıklara göre genetik arama süresince optimal kol yüksekliği 0,3 mm olduğu gösterilmektedir.



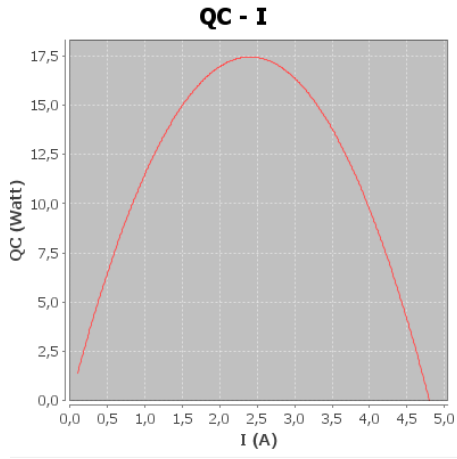
Şekil 5.10. TEGAO 1 program ile kol yüksekliğinin 50 iterasyona (turuna) göre yaklaşımı

Şekil 5.11'de 20 kez genetik operatörlerden sonra optimal kol sayısının 83 olduğu görünmektedir.

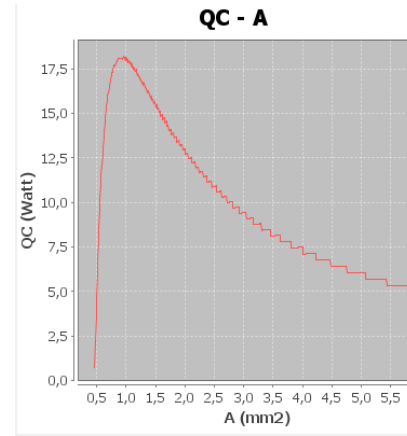


Şekil 5.11. TEGAO 1 program ile kol sayısının 50 iterasyona (turuna) göre yaklaşımı

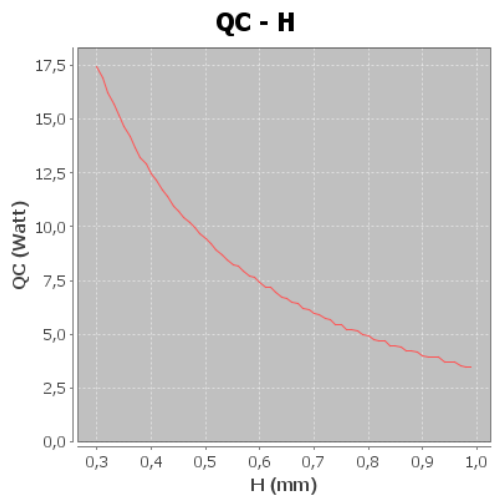
Şekil 5.12-Şekil 5.14'da Sınırlanmış bir hacimde, sabit parametreler $\Delta T = 0$, $T_H = 300K$ iken elde edilen Q_C 'nin akıma, alana ve kol yüksekliğine göre değişimi gösterilmiş ve optimal değerler verilmiştir.



Şekil 5.12. TEGAO 1 program ile Q_C 'nin akıma göre değişimi



Şekil 5.13. TEGAO 1 program ile Q_C 'nin kol alanına değişimi



Şekil 5.14. TEGAO 1 program ile Q_C 'nin kol yüksekliğine göre değişimi

TEGAO 1 programında elde edilen sonuçlar:

- Program sadece küçük hacimler için değil en büyük hacimler içinde optimizasyon işlemi yapmaktadır.
- Bir termoelemente uygulanan akım şiddeti, I_{max} 'dan büyük olduğunda Joule ısı Peltier ısından fazla olur. Bu nedenle soğuma etkisi azalır ve akımı maksimum akımın iki katına ($I = 2I_{max}$)'a ulaştığında soğutma gerçekleşmez. Akım değeri daha da arttığında, yani $I > I_{max}$ olduğundan ısısal denge bozulur. Joule ısı Peltier ısından daha fazla olur ve termoelement soğutma özelliğini kaybeder. Dolayısıyla şekil 5.12'de görüldüğü gibi I_{max} 2,46 dır. Akım I_{max} 'ın iki katına yani 4,92A'e çıktığında Q_C sıfır olur.

5.2. TEGAO 2 Programının Menüsü ve Çalışması

5.2.1. TEGAO 2'de farklı T_H 'ler için maksimum soğutma kapasitesinin hesaplanması

TEGAO 2 programı ile modülün hacmi sınırlı ve $\Delta T=0$ kabul edilerek farklı T_H sıcaklıklar için Q_C ve COP'un maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği ve akıma göre optimizasyon işlemi yapılmıştır. Böylece gerçek modülün çalışmasını etkileyen en önemli parametrelerinden T_H 'ye göre Q_C 'nin ve COP'un nasıl değiştiği tespit edilmiştir. Başka bir deyişle T_H 'nin optimizasyon üzerindeki etkisi araştırılmış ve ortaya konulmuştur.

Optimizasyon işlemi için 3 değişken parametrenin değerleri belirlenen sınırlar aralığından rastgele seçilmiştir. Seçilen bu değerler Eş. 4.1'de yerin konularak Q_{CG} değeri ve performans katsayısı hesaplanmıştır.

TEGAO 2 programında seçilen örnek uygulama için parametrelerin aralık değerleri çizelge 5.4'de gösterilmektedir. Bu parametre değerleri farklı T_H için

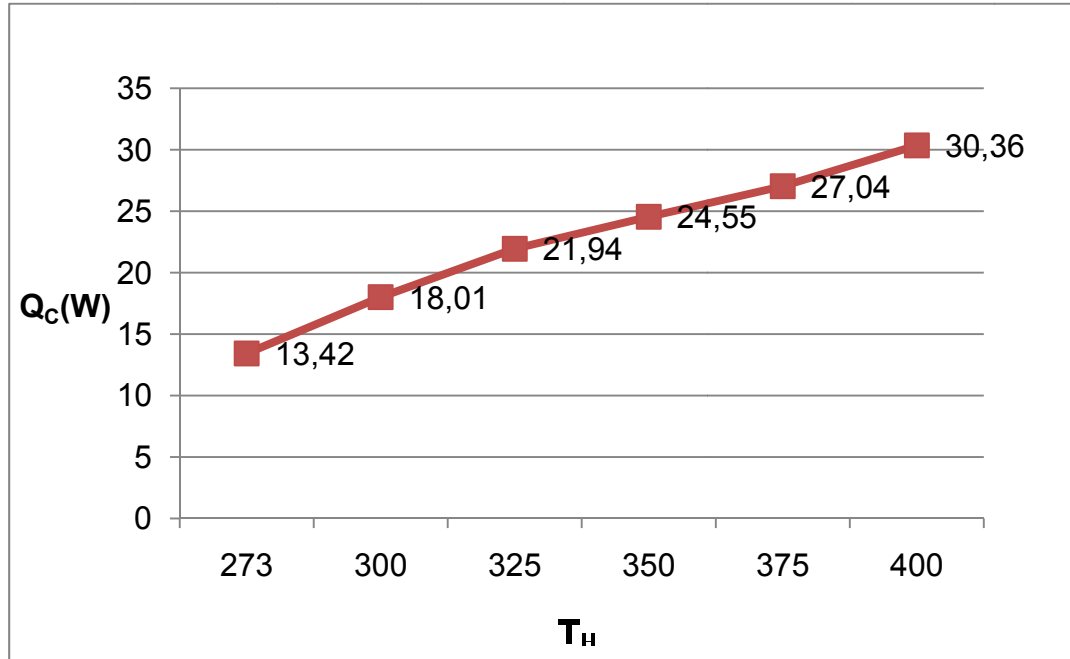
birkaç örnekte gösterilmiştir. Şekil 5.15'de görüldüğü gibi Q_C , T_H 'ye göre değişmektedir. T_H arttıkça Q_C 'nin değeri artmaktadır.

Çizelge 5.4. TEGAO 2 programında seçilen örnek uygulama kriterleri

TE.modülün Hacmi(mm ³)	Akım I(A)	Alan A(mm ²)	Yükseklik H(mm)
22 - 23	0,1 - 8	0,09 - 6	0,3 - 5

Çizelge 5.5. $\Delta T = 0$ ve farklı T_H 'ler için elde edilen optimal boyutlar

T_H	I(amper)	A(mm ²)	H(mm)	N	$Q_{CG}(W)$	COP	V(mm ³)
273	5,66	2,25	0,30	31	13,42	0,49	22,97
300	1,42	0,51	0,30	150	18,01	0,44	22,93
325	7,36	2,61	0,30	29	21,94	0,53	22,92
350	7,31	2,48	0,31	30	24,55	0,53	22,78
375	3,98	1,31	0,30	58	27,04	0,53	22,88
400	3,65	1,19	0,30	64	27,36	0,51	22,92



Şekil 5.15. Farklı T_H 'ler için Q_C değerleri

5.3. TEGAO 3 Programının Menüsü ve Çalışması

5.3.1. TEGAO 3 programında farklı T_H 'ler için maksimum soğutma kapasitesi ve maliyetin hesaplanması

TEGAO 3 programı ile modülün hacmi sınırlı ve ΔT sıfır kabul edilerek farklı T_H sıcaklıklar için Q_C ve COP'un maksimum değerlerinin ve maliyetin minimum değerinin elde edilmesi için kol alanı, kol yüksekliği ve akıma göre optimizasyon işlemi yapılmıştır.

Çizelge 5.6'da TEGAO 3 programında seçilen örnek uygulama parametrelerin aralık değerlerini göstermektedir. Bu örnekte $T_H = 300K$ alınmıştır.

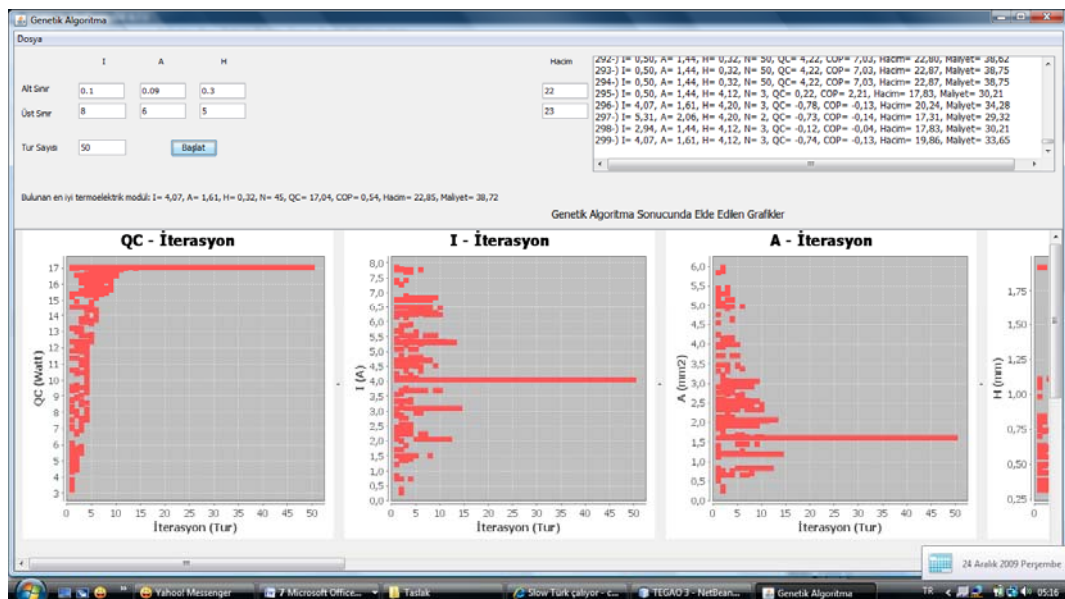
Çizelge 5.6. TEGAO 3 programında seçilen örnek uygulama kriterleri

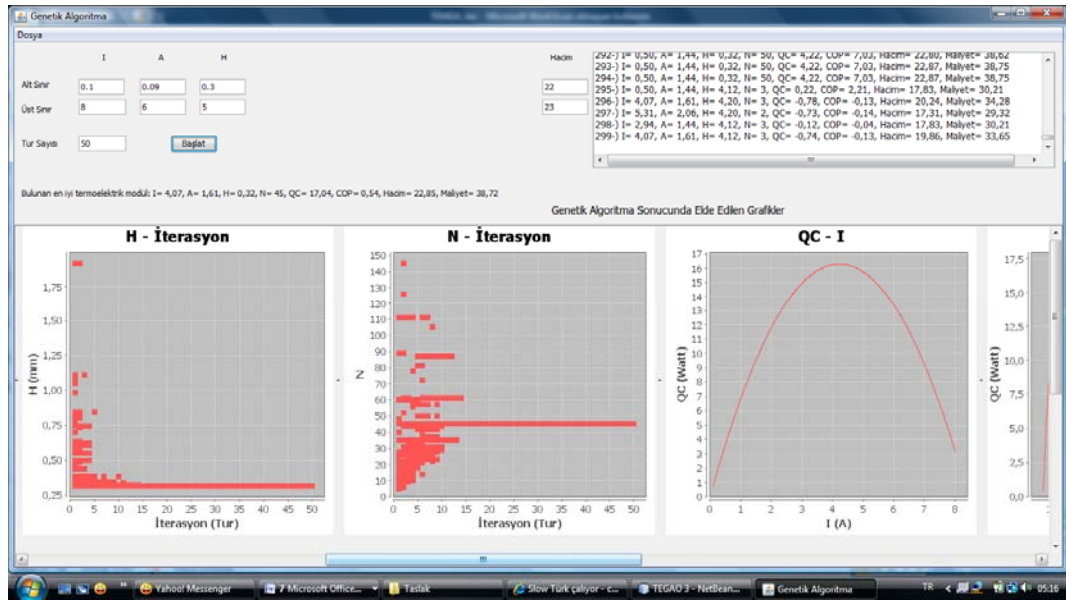
TE.modülün Hacmi(mm ³)	Akım I(A)	Alan A(mm ²)	Yükseklik H(mm)
22 - 23	0,1 - 10	0,09 - 6	0,3 - 5

Termoelektrik modülün toplam maliyeti eşitlik 4.2'den hesaplanmıştır. Toplam maliyet, maddenin toplam ağırlığının kg başına maliyet ile çarpılması şeklinde hesaplanır. Hesaplama işlemlerinde yoğunluk değeri 7,7 g/mm⁴ olan Bismuth Telluride maddesi ve bu madenin kilo başına maliyeti de 2200\$ olarak belirlenmiştir. Verilen kriterlere göre parametrelerin optimal değerleri, Q_{CG} , COP ve maliyet çizelge 5.7'deki gibi bulunmuştur. Ayrıca TEGAO 3 programın ekran görüntüleri şekil 5.16'dan 5.18'e kadar gösterilmiştir.

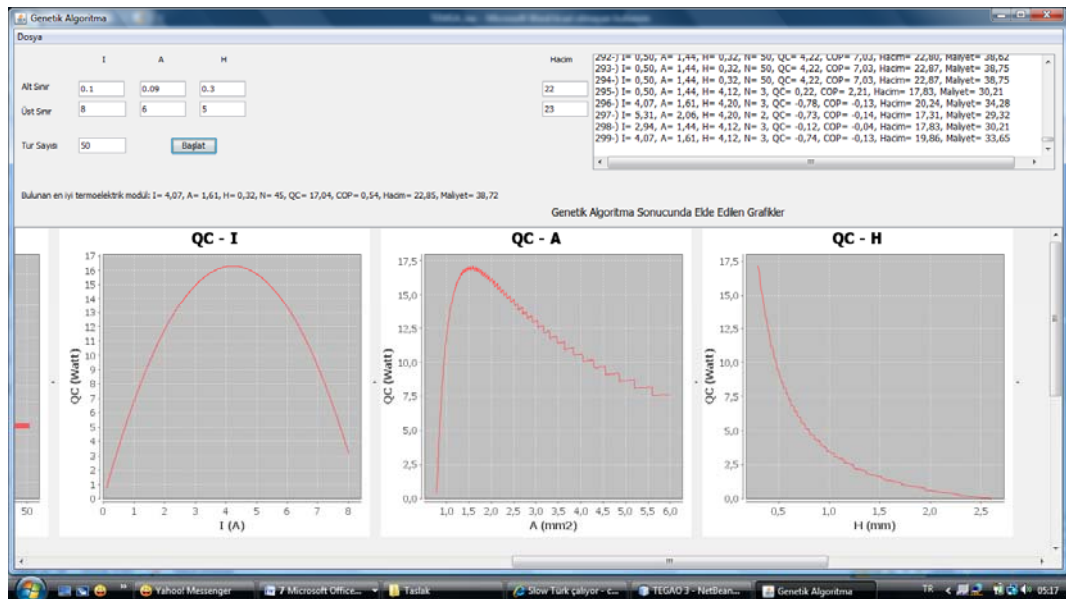
Çizelge 5.7. TEGAO 3 ile verilen kriterlere göre elde edilen parametre değerleri

I_{max} (A)	A (mm ²)	H (mm)	N	Q_{CG} (W)	COP	Maliyet (\$)	Hacim (mm ³)
4,68	1,61	0,32	45	17,04	0,54	38,72	22,85

Şekil 5.16. TEGAO 3 ile Q_C , akım ve alanın 50 iterasyona göre yaklaşımı



Şekil 5.17. TEGAO 3 programında tercih edilen aralıklar için 50 iterasyondan sonra, .sonuç .ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre ekran görüntüsü



Şekil 5.18. TEGAO 3 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre ekran görüntüsü

5.4. TEGAO 4 Programının Menüsü ve Çalışması

5.4.1. TEGAO 4 farklı ΔT 'ler ve farklı T_H 'ler için maksimum soğutma kapasitesi ve COP hesaplanması

TEGAO 4 programı ile modülün hacmi sınırlı ve T_H farklı kabul edilerek Q_C , COP'un maksimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve akıma göre optimizasyon işlemi yapılmıştır. Burada sadece T_H 'nin etkisi değil, kollar arasındaki boşluğun, Q_C ve COP üzerindeki etkisi de araştırılmıştır. Verilen kriterlere göre parametrelerin optimal değerleri, Q_{CG} , COP, T_H Çizelge 5.9'deki gibi bulunmuştur. Ayrıca TEGAO 4 programın ekran görüntüleri Şekil 5.19 – Şekil 5.21 gösterilmiştir.

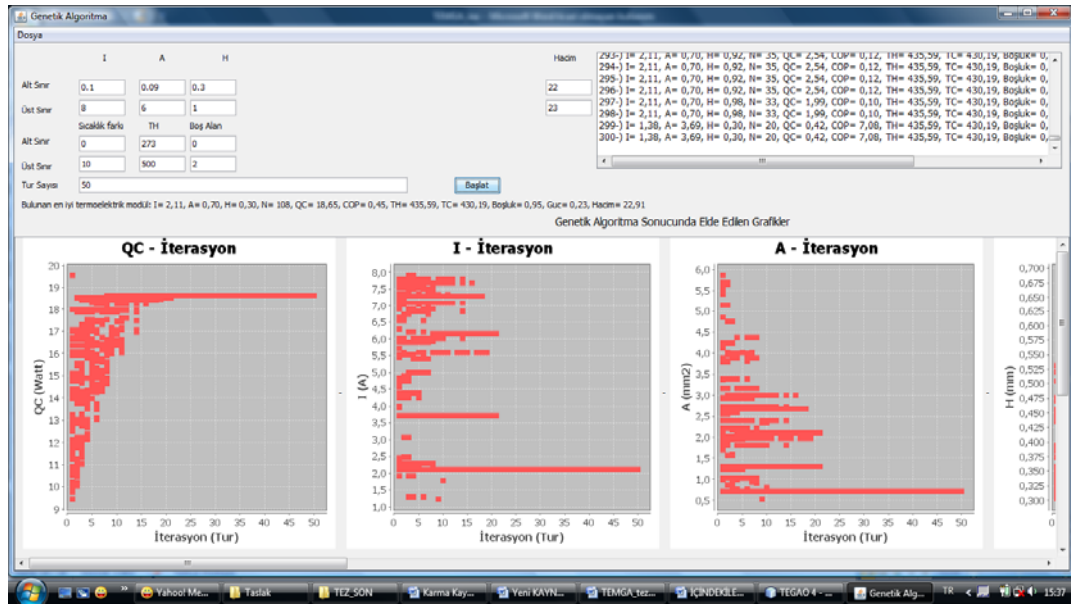
Çizelge 5.8'de TEGAO 4 programında seçilen örnek uygulama, parametrelerin aralık değerlerini göstermektedir.

Çizelge 5.8. TEGAO 5 ile verilen parametre değerleri

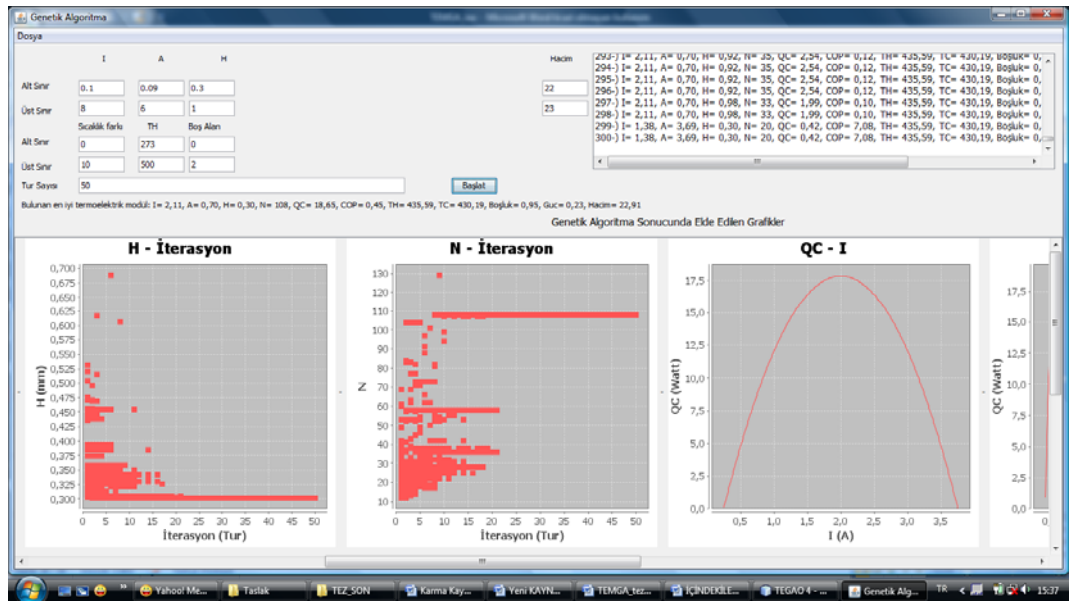
Parametre	V (mm ³)	I (amper)	A (mm ²)	H (mm)	ΔT (K)	T_H (K)	δ (mm ²)
Alt sınır	22	0,1	0,09	0,3	0	273	0
Üst sınır	23	8	6,00	5	70	750	2

Çizelge 5.9. TEGAO 5 verilen kriterlere göre elde edilen optimal parametre değerleri

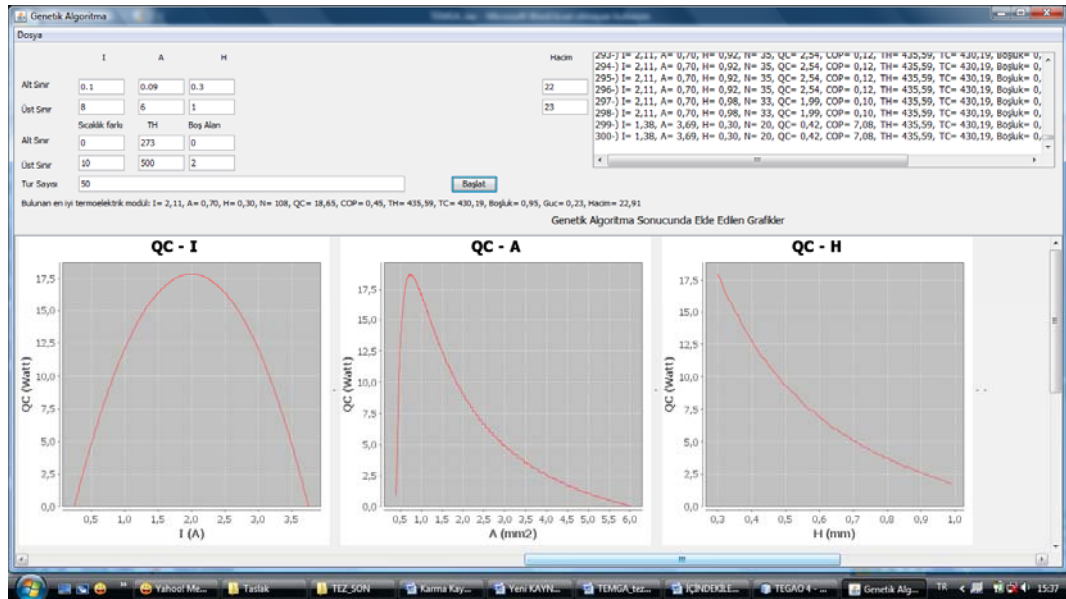
V (mm ³)	I (amper)	A (mm ²)	H (mm)	N	T_H (K)	T_C (K)	Q_{CG} (w)	COP	δ (mm ²)
22,91	2,11	0,70	0,30	108	435,99	430,19	18,65	0,45	0,95



Şekil 5.19. TEGAO 4 ile Q_c , akım ve alanın 50 iterasyona göre yaklaşımı



Şekil 5.20. TEGAO 4 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü



Şekil 5.21. TEGAO 4 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü

5.5. TEGAO 5 Programının Menüsü ve Çalışması

5.5.1. TEGAO 5 farklı ΔT 'ler için ve farklı boşluklar için maksimum soğutma kapasitesi ve minimum boşlukların hesaplanması

TEGAO 5 programı ile modülün hacmi sınırlı kabul edilerek Q_C , COP ve ΔT 'nin sıfırla maksimum değerleri arasındaki sayılarının elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve akıma göre optimizasyon işlemi yapılmıştır. Bunun sonucunda modülün Q_{Cmax} ve ΔT_{max} çalışma rejimlerinin tespit edilmesi sağlanmıştır. TEGAO 5 programının ekran görüntüleri Şekil 5.22 - Şekil 5.24'de gösterilmiştir.

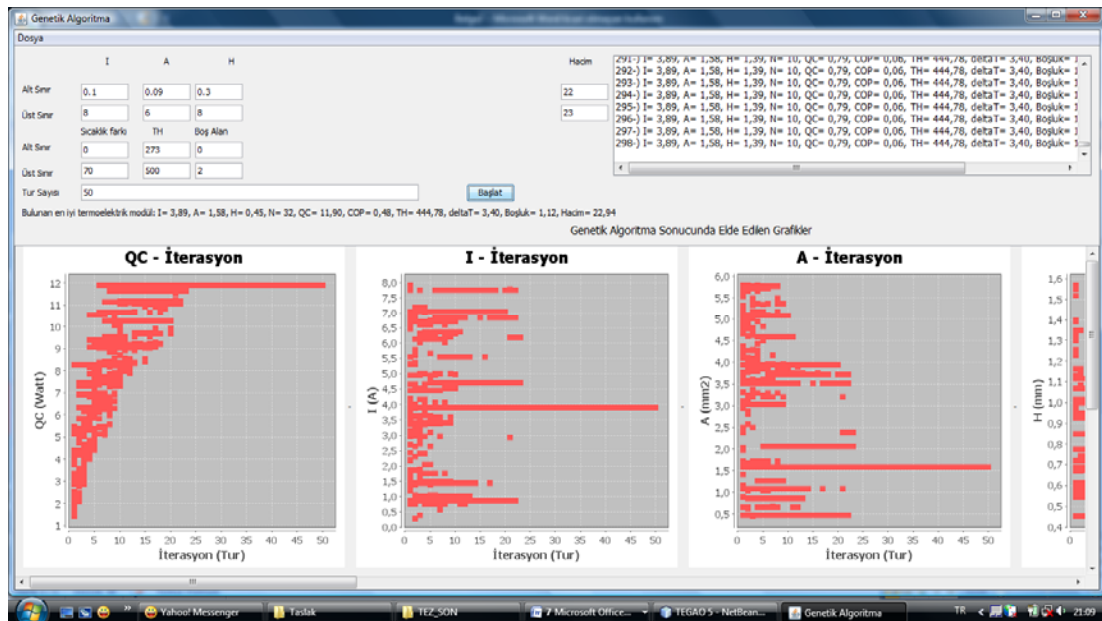
Çizelge 5.10'da TEGAO 5 programında seçilen örnek uygulama, parametrelerin aralık değerlerini göstermektedir.

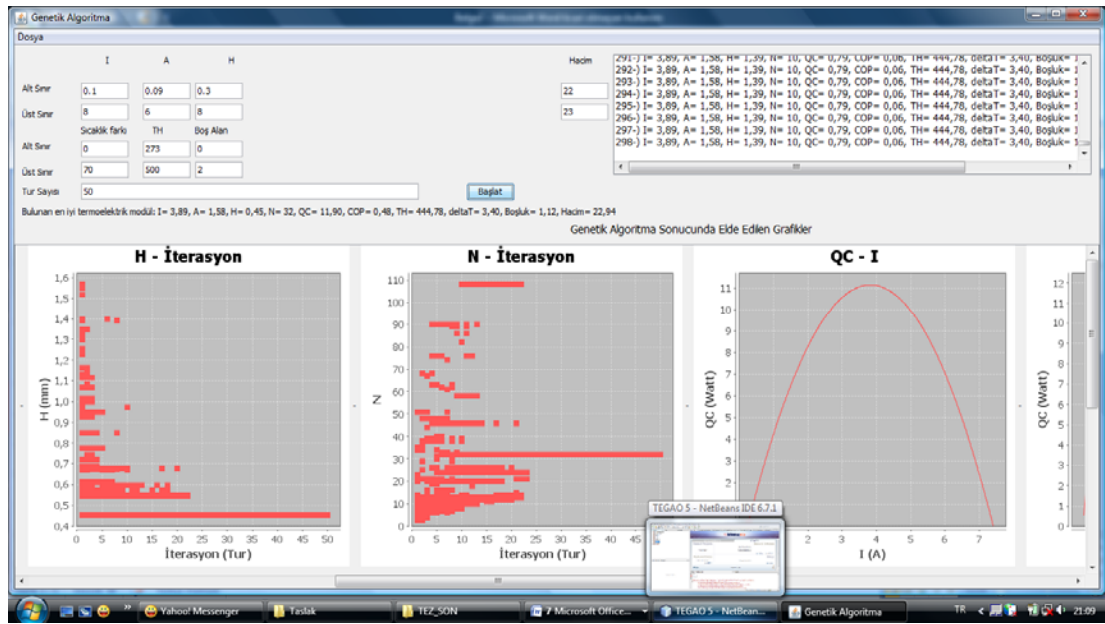
Çizelge 5.10. TEGAO 5 ile verilen parametre değerleri

Parametre	V (mm ³)	I (amper)	A (mm ²)	H (mm)	ΔT (k)	T _H (k)	δ (mm ²)
Alt sınır	22	0,1	0,09	0,3	0	273	0
üst sınır	23	8	6,	5,0	70	750	2

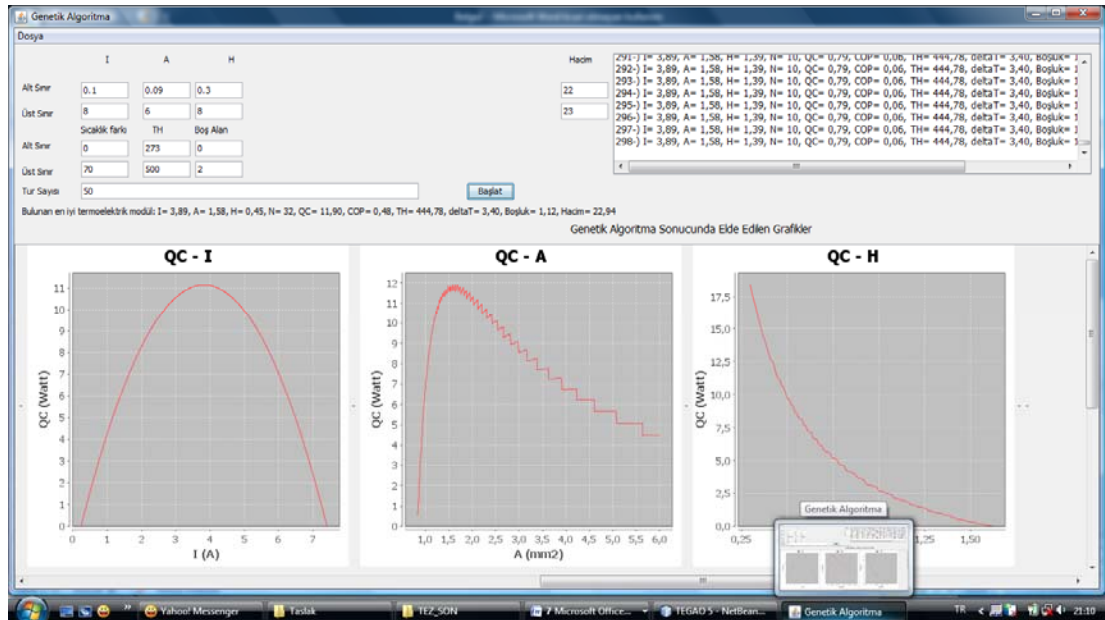
Çizelge 5.11. TEGAO 5 ile verilen kriterlere göre elde edilen optimal parametre değerleri

V (mm ³)	I (amper)	A (mm ²)	H (mm)	N	ΔT (K)	T _H (K)	δ (mm ²)	Q _{CG} (W)	COP
22,94	3,89	1,58	0,45	32	3,40	444,78	1,12	11,9	0,48

Şekil 5.22. TEGAO 5 ile Q_C, akım ve alanın 50 iterasyon göre yaklaşımı



Şekil 5.23. TEGAO 5 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü



Şekil 5.24. TEGAO 5 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü

5.6. TEGAO 6 Programının Menüsü ve Çalışması

5.6.1. TEGAO 6 farklı ΔT 'ler için ve farklı boşluklar için maksimum soğutma kapasitesi, COP ve minimum maliyetin hesaplanması

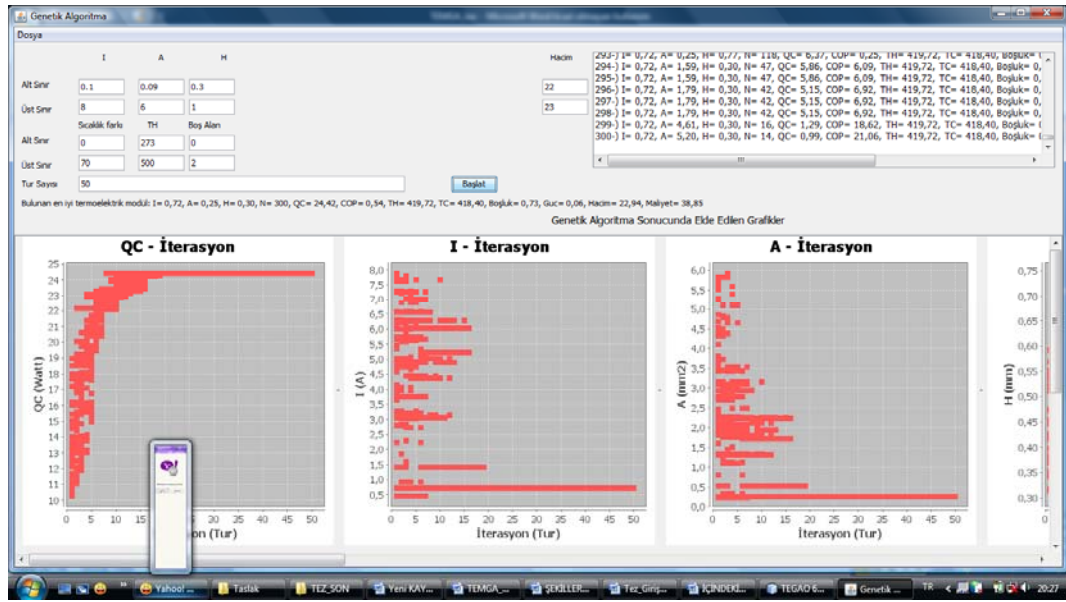
TEGAO 6 programı ile modülün hacmi sınırlı, T_H ve ΔT farklı kabul edilerek Q_C , COP maksimum ve maliyetin minimum değerlerinin elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve akıma göre optimizasyon işlemi yapılmıştır. Burada sadece T_H 'nin etkisi değil kollar arasındaki boşluğun, Q_C , P, COP ve maliyet üzerindeki etkisi araştırılmıştır. Çizelge 5.12'da TEGAO 6 programında seçilen örnek uygulama, parametrelerin aralık değerlerini göstermektedir. TEGAO 6 programın ekran görüntüleri Şekil 5.25 – Şekil 5.27'de gösterilmiştir.

Çizelge 5.12. TEGAO 6 programında verilen parametre değerleri

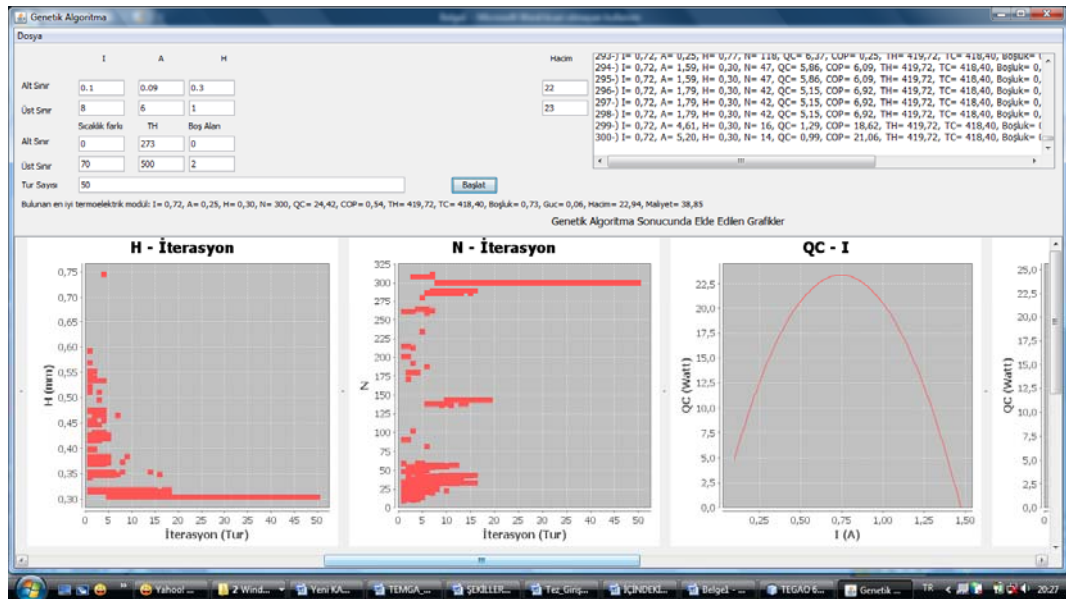
Parametre	Hacim (mm ³)	I (amper)	A (mm ²)	H (mm)	ΔT (k)	T_H (k)	δ (mm ³)
Alt sınır	22	0,1	0,09	0,3	0	273	0
Üst sınır	23	8	6	5	70	750	2

Çizelge 5.13. TEGAO 6 programında verilen kriterlere göre elde edilen optimal parametre değerleri

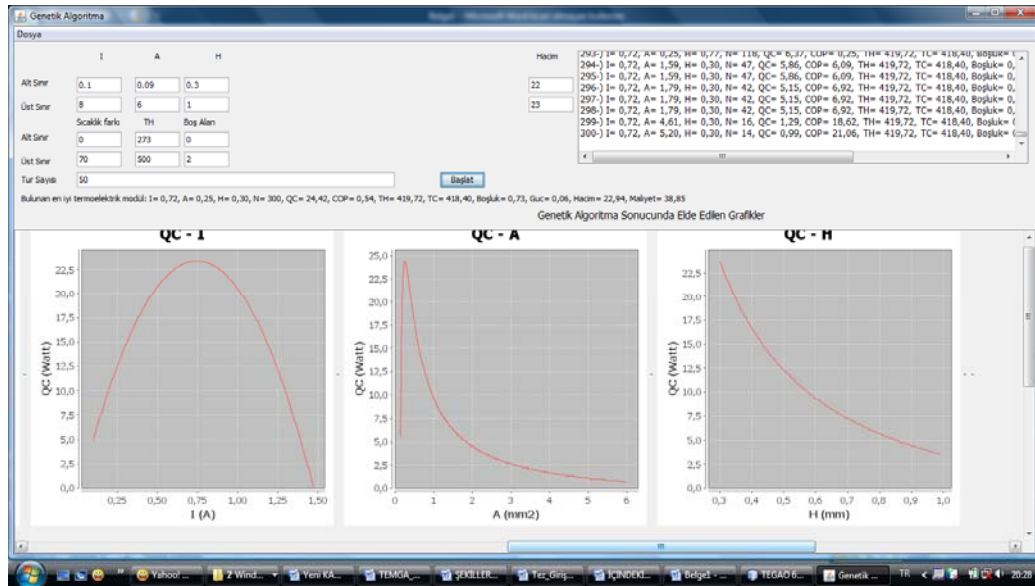
I	A (mm ²)	H (mm)	N	T_H (K)	TC (K)	δ (mm ³)	P (K)	Q_{CG} (W)	COP	Maliyet (\$)	V (mm ³)
0,72	0,25	0,3	300	410,72	418,40	0,73	0,06	24,42	0,54	38,85	22,94



Şekil 5.25. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, maksimum soğutma kapasitesi, maksimum akım ve optimal kol alanının görüntüsü



Şekil 5.26. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, sonuç ekran görüntüsü, optimal kol yüksekliği, kol sayısı ve akımın soğutma kapasitesine göre görüntüsü



Şekil 5.27. TEGAO 6 programında tercih edilen aralıklar için 50 iterasyondan sonra, bulunan maksimum akım, optimal kol alanı ve optimal kol yüksekliğinin soğutma kapasitesine göre görüntüsü

5.7. TEGAO İLE ELDE EDİLEN SONUÇLARIN MELCOR SONUÇLARI İLE KARŞILAŞTIRMASI

Genetik algoritmayla çeşitli modüler için elde edilen maksimum soğutma kapasitelerinin ne kadar doğru hesaplandığını araştırmıştır. Bunun için Melcor'un ürettiği CP serisine ait 36 adet termoelektrik modülleri maksimum soğutma kapasiteleri önce eşitlik 4.1 ve 4.2 den hesaplanmış ve elde edilen sonuçlar genetik algoritma yöntemi ile elde edilenlerle karşılaştırılmıştır.

Bunun için eşitlik 5.2 $\Delta Q_C(W)$, Melcor firmasının ürettiği TE. modüllerinin soğutma kapasitesi (Q_{CM}) ile genetik algoritma yöntemiyle elde edilen soğutma kapasitesinin (Q_{CG}) farkı;

$$\Delta Q_C(w) = |Q_{CG} - Q_{CM}| \quad (5.2)$$

ve farklılık yüzdesi (ε):

$$\varepsilon = \left| 1 - \frac{Q_{CG}}{Q_{CM}} \right| * \%100 \quad (5.3)$$

olarak hesaplanmıştır. Çizelge 5.14'de. Melcor'un 36 adet TE modüller için katalog bilgileri ile TEGAO programında hesaplanan sonuçlar karşılaştırılmıştır.

Çizelge 5.14. Melcor ile TEGAO programının karşılaştırılma sonuçları

Melcor serileri	I (amper)	A (mm ²)	H (mm)	G (mm)	N	Q _{CM} (w)	Q _{CG} (w)	ΔQ _c (W)	□%
CP 0.8-7-06L	2,1	1,43	3,4	0,42	7	1,0	0,97	0,03	3
CP 0.8-17-06L	2,1	1,43	3,4	0,42	17	2,4	2,36	0,04	2
CP 0.8-31-06L	2,1	1,43	3,4	0,42	31	4,4	4,3	0,10	2
CP 0.8-63-06L	2,1	1,43	3,4	0,42	63	9	8,74	0,26	3
CP 0.8-71-06L	2,1	1,43	3,4	0,42	71	10,1	9,85	0,25	2
CP 0.8-127-06L	2,1	1,43	3,4	0,42	127	18,1	17,62	0,48	3
CP 1.0-7-08L	2,5	2,0	4,0	0,50	7	1,2	1,27	0,07	6
CP 1.0-17-08L	2,5	2,0	4,0	0,50	17	2,9	3,07	0,17	6
CP 1.0-31-08L	2,5	2,0	4,0	0,50	31	5,3	5,61	0,31	6
CP 1.0-63-08L	2,5	2,0	4,0	0,50	63	10,6	11,39	0,79	7
CP 1.0-71-08L	2,5	2,0	4,0	0,50	71	12	12,84	0,84	7
CP 1.0-127-08L	2,5	2,0	4,0	0,50	127	21,4	22,97	1,57	7
CP 1.0-7-06L	3,0	2,2	3,6	0,61	7	1,4	1,48	0,08	6
CP 1.0-17-06L	3,0	2,2	3,6	0,61	17	3,4	3,6	0,20	6
CP 1.0-31-06L	3,0	2,2	3,6	0,61	31	6,3	6,57	0,27	4
CP 1.0-63-06L	3,0	2,2	3,6	0,61	63	12,7	13,35	0,65	5
CP 1.0-71-06L	3,0	2,2	3,6	0,61	71	14,4	15,05	0,65	5
CP 1.0-127-06L	3,0	2,2	3,6	0,61	127	25,7	26,92	1,22	5
CP 1.0-7-05L	3,9	2,5	3,2	0,79	7	1,8	1,78	0,02	4
CP 1.0-17-05L	3,9	2,5	3,2	0,79	17	4,5	4,31	0,19	7
CP 1.0-31-05L	3,9	2,5	3,2	0,79	31	8,2	7,86	0,34	6
CP 1.0-63-05L	3,9	2,5	3,2	0,79	63	16,6	15,98	0,62	6
CP 1.0-71-05L	3,9	2,5	3,2	0,79	71	18,7	18,01	0,69	6
CP 1.0-127-05L	3,9	2,5	3,2	0,79	127	33,4	32,21	1,19	6
CP 1.4-11-06L	6,0	4,5	3,8	1,18	11	4,4	4,53	0,13	3
CP 1.4-17-06L	6,0	4,5	3,8	1,18	16	6,9	6,59	0,13	4
CP 1.4-31-06L	6,0	4,5	3,8	1,18	30	12,5	12,37	0,23	1
CP 1.4-35-06L	6,0	4,5	3,8	1,18	35	14,2	14,43	0,23	2
CP 1.4-71-06L	6,0	4,5	3,8	1,18	70	28,7	28,85	0,15	1

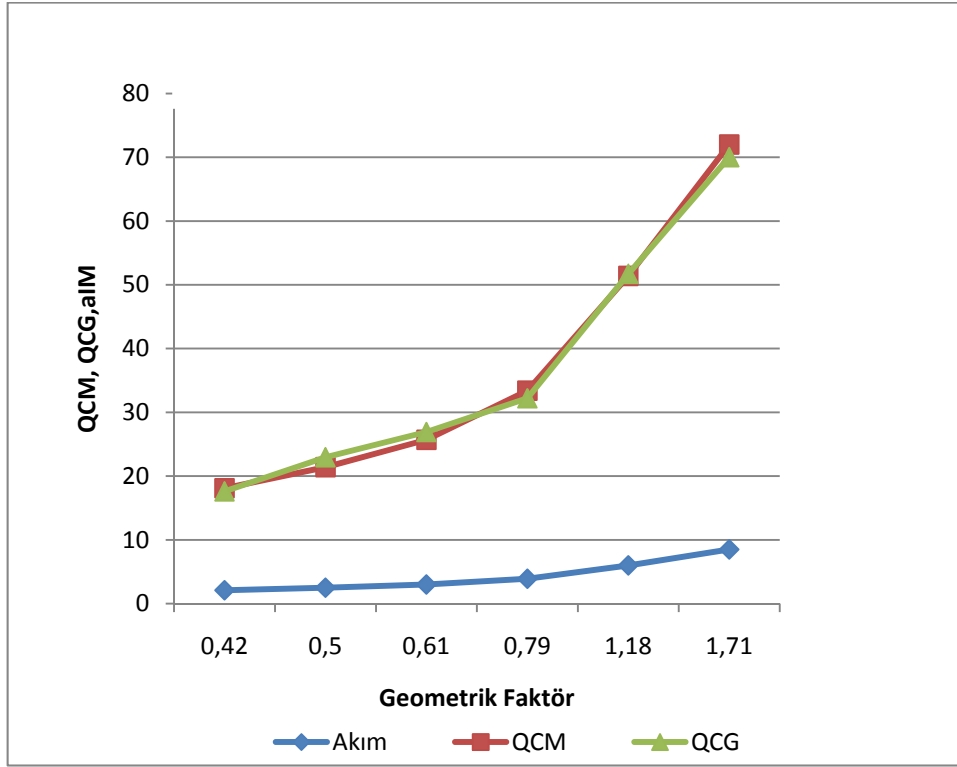
Çizelge 5.14. (Devamı) Melcor ile TEGAO programının karşılaştırılma sonuçları

Melcor serileri	I (amper)	A (mm ²)	H (mm)	G (mm)	N	Q _{CM} (w)	Q _{CG} (w)	ΔQ _c (W)	□%
CP 1.4-127-06L	6,0	4,5	3,8	1,18	127	51,4	51,71	0,31	1
CP 1.4-11-045L	8,5	5,6	3,3	1,71	11	6,0	6,02	0,02	0
CP 1.4-17-045L	8,5	5,6	3,3	1,71	17	9,2	9,3	0,10	1
CP 1.4-31-045L	8,5	5,6	3,3	1,71	31	16,8	16,96	0,16	1
CP 1.4-35-045L	8,5	5,6	3,3	1,71	35	19,0	19,14	0,14	1
CP 1.4-71-045L	8,5	5,6	3,3	1,71	71	38,5	38,83	0,33	1
CP 1.4-127-45L	8,5	5,6	3,3	1,71	128	72,0	70,01	1,99	3

Çizelge 5.15'de iki yöntemle elde edilen modüllerin soğutma kapasitelerinin geometrik faktöre göre nasıl değiştiğini gösterilmiştir. Ayrıca yapılan kıyaslamalar Şekil 5.28 da verilmiştir.

Çizelge 5.15. Melcor ve TEGAO programı ile elde edilen modüllerin soğutma kapasitelerinin geometrik faktöre göre dağılımı

G (mm)	I (amper)	Q _{CM} (W)	Q _{CG} (W)
0,42	2,1	18,1	17,62
0,50	2,5	21,4	22,97
0,61	3,0	25,7	26,92
0,79	3,9	33,4	32,21
1,18	6,0	51,4	51,71
1,71	8,5	72,0	70,01



Şekil 5.28. Q_{CM} , Q_{CG} ve akımın geometrik faktöre göre değişimi

Şekil 5.28 den görüldüğü gibi iki farklı yöntemle elde edilen Q_{CM} , Q_{CG} ve akım değerlerinin birbirleriyle örtüşmektedir.

6. SONUÇ VE ÖNERİLER

Bir termoelektrik cihazın tasarımında, kullanılacak termoelektrik modülün (TE) doğru seçilmesi çok önemlidir. Modüllerin seçiminde ise en önemli faktörler TE modülün soğutma kapasitesinin, performans katsayısının ve yüzeyler arasındaki sıcaklık farkının yüksek ve modül maliyetinin düşük olmasıdır. Bu çalışmada üretime geçilmeden termoelektrik modüllerin kollarının boyutlarının optimize ederek daha yüksek soğutma kapasiteli ve performanslı ve daha düşük maliyetli termoelektrik modüllerin tasarım programı elde edilmiştir. Java programlama dili kullanarak NetBeans 7.6.1. ortamında genetik algoritma yöntem ile 6 adet TEGAO optimizasyon programları yazılmıştır. Bu programlarla modülün toplam hacmini sınırlı tutarak, boşlukların toplam hacmi; kol alanı, kol yüksekliği, sıcak yüzeyin sıcaklığı ve kollardan geçen akıma göre optimizasyon işlemleri gerçekleştirilmiştir.

Başlangıç popülasyon olarak 300 adet termoelektrik modül seçilmiştir. Genetik algoritma yöntemi ile optimizasyon işlemi yapılarak TE modülün tüm parametrelerin optimal değerleri ve maliyeti elde edilmiştir. Ayrıca Melcor firmasının üretmiş olduğu 36 adet termoelektrik modülün her biri için geometrik parametrelerini dikkate alarak termoelektrik modüller için maksimum soğutma kapasitesi elde edilmiştir. Elde edilen sonuçlar ile Melcor'un termoelektrik modülleri için öngördüğü soğutma kapasiteleriyle karşılaştırılmış ve farklılık yüzdeleri hesaplanmıştır. Farklılık yüzdelerinin çok düşük çıkması optimizasyonun doğru yapıldığını göstermiştir. Bu optimizasyon işleminin sonucunda gerçekleştirilmiş maksimum soğutma kapasitesine sahip modüller, Melcor firmasının katalogundaki modüllerden çok daha fazla seçenek sunmaktadır.

TEGAO programlarının elde edilen sonucular:

TEGAO 1 programında sınırlı hacimde ΔT 'yi sıfır alarak ve T_H 'yi 300K kabul edilerek Q_{Cmax} , COP ve optimal kol alanı, kol yüksekliği, kol sayısı ve akım elde edilmiştir.

TEGAO 2 programında sınırlı hacimde ΔT 'yi sıfır ve sıfırdan farklı alarak ve T_H 'yi 300K kabul edilerek farklı sıcaklıklar için maksimum soğutma kapasitesi, COP ve optimal kol alanı, kol yüksekliği, kol sayısı ve maksimum akım elde edilmiştir. Böylece gerçek modülün çalışmasını etkileyen en önemli parametrelerinden T_H 'ye göre Q_C 'nin ve COP'un nasıl değiştiği tespit edilmiştir. Başka bir deyişle T_H 'nin optimizasyon üzerindeki etkisi araştırılmış ve T_H arttıkça soğutma kapasitesinin arttığı gösterilmiştir.

TEGAO 3 programında modülün hacmini sınırlayarak, maksimum soğutma kapasitesi, akım, COP ve optimal kol alanı, kol yüksekliği, kol sayısı elde edilmesinin yanı sıra minimum maliyetle elde edilmesi sağlanmıştır.

TEGAO 4 programında modülün hacmini sınırlayarak ve T_H farklı kabul edilerek Q_C ve COP'un maksimum değerleri ve kol alanı, kol yüksekliği, kol sayısı, kollar arasındaki boşluk ve akımın optimal değerleri elde edilmiştir. Burada sadece T_H 'nin etkisi değil, kollar arasındaki alanın da etkisi araştırılmıştır ve δ 'n daralmasıyla Q_C , P ve COP'un arttığı görülmektedir.

TEGAO 5 programında ile modülün hacmi sınırlı kabul edilerek soğutma kapasitesi, performans katsayısı ve ΔT 'nin sıfırla maksimum değerleri arasındaki sayılarının elde edilmesi için kol alanı, kol yüksekliği, kollar arasındaki boşluk ve akıma göre optimizasyon işlemi yapılmıştır. Bunun sonucunda modülün Q_C 'nin ΔT 'ye göre nasıl değiştiği gösterilmiştir.

TEGAO 6 programı ile modülün hacmi sınırlı, T_H ve ΔT farklı kabul edilerek Q_C ve COP'un maksimum ve maliyetin minimum değeri elde edilmiştir.

Geliştirilen optimizasyon yöntemiyle TE modülün çeşitli modüllerini kullanarak optimizasyon yapılabilir. Ayrıca TE modülün üretiminde kullanılan teknolojik faktörlerin hesaba katarak TEGAO programları ile Optimizasyon işlemi yapılabilir.

KAYNAKLAR

1. Harman, T. C., "Special techniques for measurement of thermoelectric properties", **J. Appl. Phys.**, 29: 1373 - 1379 (1958).
2. Ioffe, A. F., "Polucenie holoda poluprovodnikovie termoelementi", **Moskov Pres**; 14-24(1960).
3. Heylen, A., "Figure of merit determination of thermoelectric modules", **Energ. Convers. Manage**, 15: 65 – 70 (1975).
4. Buist, R. J., "Methodology for testing thermoelectric materials and devices", Rowe D. M, editors. CRC handbook of thermoelectrics, florida: **CRC Pres. Inc.**; 189-209 (1995).
5. Min, G., Rowe, D. M., "A novel principle allowing rapid and accurate measurement of a dimensionless thermoelectric figure of merit", **Measur. of Scie. And Technol.**, 12: 1261 - 1262 (2001).
6. Xuan, X. C., Ng, K.C., Yap, C., Chua, H. T., "A general model for studying effects of interface layers on thermoelectric devices performance", **Intern. J. of Heat and Mass Trans.**, 45: 5159 – 5170 (2002).
7. Xuan, X. C., "Investigation of thermal contact effect on thermoelectric coolers", **Energ. Convers. Manage**, 4: 399 - 410 (2003).
8. Cheng, Y.-H., Shih, C., "Maximizing the cooling capacity and COP of two-stage thermoelectric coolers through genetic algorithm", **Applied Thermal Engineering**, 26: 937 - 947 (2006).
9. Yavuz, A. H., "Bulanık mantık denetimli termoelektrik beyin soğutucusu", **Gazi Üniversitesi, Bilişim Enstitüsü, Elektronik ve Bilgisayar Eğitimi**, Doktora Tezi, Ankara, 11 – 30 (2009).
10. Fidan, U., Ahiska, R., Güler, N.F., "Mikrodenetleyici kontrollü taşınabilir termoelektrik tıp kiti", **Elektrik - Elektronik - Bilgisayar Mühendisliği Sempozyumu**, Bursa, 8-12: 255 - 259 (2000).
11. Huang, B.J. , Chin, C.J., Duang, C.L. "A design method of thermoelectric cooler", **Taiwan International Journal of Refrigeration**, 23: 208-218 (2000).
12. Cheng, Y.-H., Lin, W.-K., "Geometric optimization of thermoelectric coolers in a confined volume using genetic algorithms", **Applied Thermal Engineering**, 25: 2983 – 2997 (2005).

13. Cheng, Y.-H., Shih, C., "Maximizing the cooling capacity and COP of two-stage thermoelectric coolers through genetic algorithm", **Applied Thermal Engineering**, 26: 937 - 947 (2006).
14. İşçi, Ö., Korukoğlu, S., "Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama", **Yönetim ve Ekonomi**, Celal Bayar Üniversitesi, 10: 2 (2003).
15. Olinsky, A.D., Quinn, J. T., "A genetic algorithm approach to nonlinear least squares estimation", **Int. J. Math. Educ. Sci. Technol.**, 35(2): 207 – 217(2004).
16. Karaboğa, D., "Yapay zeka optimizasyon algoritmaları", **Atlas Yayınevi**, İstanbul, 5-7 (2004).
17. Gen, M., Cheng, R., "Genetic algorithms and engineering design", **John Wiley & Sons Inc.**, New York 1-7 (1997).
18. Holland, J., "Adaptation in natural and artificial systems", **University of Michigan Press, Ann Arbor**, 112-120 (1975).
19. Goldberg, D. E., Genetic algorithms in search optimization and machine learning, **Addison-Wesley Publishing Company Inc.**, Massachusetts, 1 - 20, 221, 412 (1989).
20. Blickle, T., Thiele, L." A comparison of selection schemes used in genetic algorithms", **TIK-Report**, 11: 1 - 67 (1995).
21. Mitchell, M., "An introduction to genetic algorithms", **MIT Press**, Massachusetts, 205 - 213 (1996).
22. Michalewicz, Z., "Genetic algorithms + data structures = evolution programs", **Springer-Verlag**, 340 - 353 (1994).
23. Internet: Melcor, "Thermoelectric handbook", <http://www.melcor.com>, (2000).
24. Cheng, Yi-H., Shih, C., "An intelligent code to design thermoelectric heat pumps" **National Tsing Hua University Department of Engineering and System** , Taiwan, 223 - 260 (2007).
25. Internet: TETECH, "Custom and standart thermoelectric applications", <http://www.tetech.com> (2009).
26. Internet: Türkiye Bilimsel ve Teknoloji Araştırma Kurumu Web Sitesi, <http://www.biltek.tubitak.gov.tr/gelisim/elektronik/dosyalar/26/termoelektrik.pdf> (2007).

27. Salimi Altan, M., "Termoelektrik sistemlerin analiz programı", Yüksek Lisans Tezi, **Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik-Bilgisayar Eğitimi**, Ankara, 19 - 22 (2001).
28. Ahıska, R., Güler, E., Acar, S., Mehmet, K., "Thermoelectric characterization of n-type $(\text{Bi}_2\text{Te}_3)\text{Se}_3$ semiconductors in a temperature range 11– 373 K", **G.U. Journal of Science**, 18:(3), 187 - 191(2005).
29. Dişlitaş, S., "Mikrodenetleyici kontrollü jeotermal termoelektrik jeneratör tasarımı ve uygulaması, Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 11 - 23 (2002).
30. Ahıska, R., Dişlitaş, S., "Microcontroller Based Thermoelectric Generator Application", **G.U. Journal of Science**, 19:(2), 135 - 141 (2006).
31. Min, G., Rowe, D.M., "Improved model for calculating the coefficient of performance of a Peltier module", **Energy Conversion & Management** 41: 163 - 171 (2000).
32. Rowe, D.M., "CRC Handbook of thermoelectrics", **CRS Press LLC**, New York, 211 (1994).
33. Ahıska, R., Ahıska, K., "New method for investigation of parameters of real thermoelectric modules", **Energy Conversion and Management**, 51: 338 - 345 (2009).
34. Ahıska, R., Ahıska, G., Ahıska, K., "Analysis of a new method for measurement of parameters of real thermoelectric module employed in medical cooler for renal hypothermia", **Instrumentation Science & Technology**, 37 (1) : 102 - 123 (2009).
35. Ahıska, R., "Comparing analysis of a new method for measurement of parameters of real thermoelectric modules", **XIII International Forum on Thermoelectricity**, Ukraine, 28 - 36 (2009).
36. Ahıska, R., "The study of thermoemf effect on exit parameters of thermoelectric modules", **International Conference "Fizika – 2005"**, Azerbaijan, 144 - 148 (2005).
37. Ahıska, R., "New method for studying parameters of a thermopile", **Journal of Thermoelectricity**, 3 : 71 - 82 (2008).
38. Ahıska, R., "New method for investigation of dynamic parameters of thermoelectric modules", **Turkish Journal of Electrical Engineering**, 15 (1) : 51 - 65 (2007).

39. Ahıska, R., Ciylan, B., Savaş., Y., Güler, İ., “Standart termoelektrik modülün Z parametresinin ölçülmesi için yeni yöntem ve yeni sistem”, **Gazi Üniversitesi Mühendislik Fakültesi Dergisi**, 19 (4) : 467 - 473 (2004).
40. Dişlitaş, S., Ahıska, R., Yanmaz, H., “Termoelektrik Modülün Dinamik Çıkış Parametrelerinin Elde Edilmesi İçin Yeni Bir Algortma”, **Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu**, 471 - 475 (2008).
41. Sağıroğlu, Ş., Beşdok, E., Eler, M., “Mühendislikte yapay zeka uygulamaları – I., Yapay Sinir Ağları”, **Ufuk Kitap Kirtasiye - Yayıncılık Tic. Ltd. Şti.**, Kayseri, 4 - 21 (2003).
42. Elmas, Ç., “Yapay sinir Ağları (kuram, mimari, eğitim, uygulama),” **Seçkin Yayıncılık San.ve Tic.A.Ş.**, Ankara, 21 - 25 (2003).
43. Bal, H., “Optimizasyon teknikleri”, **Gazi Üniversitesi Yayınları**, Ankara, 2-6 (1995).
44. Carlos, A., Coello, C., “An updated survey of ga-based multiobjective optimization techniques”, Laboratio Nacional de Informática Avanzada **ACM Computing Surveys**, 32: 2 (2000).
45. Ulusoy, G., “Proje planlamada kaynak kısıtlı çizelgeleme”, **Sabancı Üniversitesi Mühendislik Ve Doğa Bilimleri Fakültesi Dergisi**, 8:23-29 (2002).
46. Blickle, T., Thiele, L., “A comparison of selection schemes used in evolutionary algorithms”, **Massachusetts Institute of Technology, Evolutionary Computation** 4(4): 361 - 394 (1997).
47. Haupt, R. L., Haput, S., ”Practical genetic algorithms” **A Wiley-Interscience Publication**, New York, 17 (1998).
48. Ponce-Ortega, J. M., Serna-Gonzalez, M., Jimenez-Gutierrez, A., “Use of Genetic Algorithms For The Optimal Design of Shell-and-Tube Heat Exchangers”, **Applied Thermal Engineering**, 29, 203 - 209 (2009).
49. Llor’a, X., “Data-Intensive Computing for Competent Genetic Algorithms: A Pilot Study using Meandre” , **Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign**, IlliGAL University, February, 118 – 143 (2009).
50. Garg, D.P., Kumar, M., “Optimization Techniques Appliedto Multiple Manipulators for Path Planning And Torque Minimization”, **Engineering Applications of Artificial Intelligence**, 15, 241 – 252 (2002).

51. Bridges, C.L, Goldberg,D.E., “An analysis of reproduction and in a binary coded genetic algorithm”, ***Proceedings of the Second International Conference on Genetic Algorithms***, 9 - 13 (1987).
52. Davis, L., “Handbook of genetic algorithms”, ***Van Nostrand Reinhold, NY***, 385 - 392 (1991).
53. Reeves, C. “Modern heuristic methods for combinatorial problems”, ***Blackwell Scientific Publications***, 320 - 328 (1993).
54. Painton, L. A., Campbell, J. E., “Genetic algorithms in optimization of system reliability” ***IEEE Transactions on Reliability***, 44, 172 - 178 (1995).
55. Michalewicz, Z., Fogel, D. B., “How to solve it: modern heuristics” ***Second, Revised and Extended Edition***, 117-121 (2004).
56. Tomassini, M., “Evolutionary algorithms”, Lecture Notes in Computer Science, ***Springer-Verlag***, Berlin, 19–47 (1996).
57. Srinivas, M., Patnaik,L.M., “Genetic algorithms: A Survey”, ***Computer***, 27(6): 17-26 (1994).
58. Srinivas, M., Patnaik,L.M., “Adaptive probabilities of crossover and mutation in genetic algorithms”, ***IEEE Transactions on Systems, Man and Cybernetics***, 24: 4 (1994).
59. Man, K.F., Tang, K.S., Kwong, S. “Genetic algorithms: concepts and applications”, ***IEEE Transactions on Industrial Electronics***, 43(5): 519 - 533 (1996).
60. Grefenstette, J.J., “Optimization of control parameters for genetic algorithm”, ***IEEE Transactions on systems and Cybernetics***, 16(1): 122 – 128 (1986).
61. Forrest, S., “Genetic algorithms: principles of nature selection applied to computation”, ***Science***, 261, 872-878 (1993).
62. Tomassini, M., “A survey of genetic algorithms”, ***Annual Reviews of Computations Physics***, 3: 87-118 (1995).
63. Liepins, G.E., Hillard, M.R., “Genetic algorithms: foundation and applications”, ***Annals of operations Research***, 2: 31 - 58 (1989).
64. Gonzales, E. L., Fernandez, M.A.R., “Genetic optimisator of a fuzzy distribution model” ***Inter National Journal of Physical Distribution and Logistics Management***, 30(7/8): 681 - 696 (2000).

EKLER

Ek1 TEGAO programın örnek kodları

```

*/genalgoritma;

import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JDialog;
import javax.swing.JFrame;
import java.awt.Dimension;
import java.util.Vector;
import java.util.Random;
import javax.swing.JTextArea;

public class GenAlgoritmaView extends FrameView {

    public void buyut() {
        int screenWidth =
java.awt.Toolkit.getDefaultToolkit().getScreenSize().width;
        int screenHeight =
java.awt.Toolkit.getDefaultToolkit().getScreenSize().height;
        getFrame().setBounds(0, 0, screenWidth, screenHeight - 35);
        getFrame().setSize(screenWidth, screenHeight - 35);
        getFrame().doLayout();
        getFrame().validate();
    }

    public GenAlgoritmaView(SingleFrameApplication app) {
        super(app);

        initComponents();

        ResourceMap resourceMap = getResourceMap();
        int messageTimeout =
resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate =

```

```

    }
});

* initialize the form.
* WARNING: Do NOT modify this code. The content of this method is
* always regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    jButton1 = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    iiAlt = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    turKutu = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
    jScrollPane2 = new javax.swing.JScrollPane();
    jPanel1 = new javax.swing.JPanel();
    graf4 = new javax.swing.JLabel();
    graf3 = new javax.swing.JLabel();
    graf2 = new javax.swing.JLabel();
    graf1 = new javax.swing.JLabel();
    graf5 = new javax.swing.JLabel();
    graf6 = new javax.swing.JLabel();
    graf7 = new javax.swing.JLabel();
    graf8 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    hacUst = new javax.swing.JTextField();
    hacAlt = new javax.swing.JTextField();
    menuBar = new javax.swing.JMenuBar();
    }
});
jLabel2.setText(resourceMap.getString("jLabel2.text")); // NOI18N
jLabel2.setName("jLabel2"); // NOI18N
jLabel3.setText(resourceMap.getString("jLabel3.text")); // NOI18N
jLabel3.setName("jLabel3"); // NOI18N
jLabel4.setText(resourceMap.getString("jLabel4.text")); // NOI18N
jLabel4.setName("jLabel4"); // NOI18N
jLabel5.setText(resourceMap.getString("jLabel5.text")); // NOI18N

```



```

jLabel5.setName("jLabel5"); // NOI18N
iiAlt.setText(resourceMap.getString("iiAlt.text")); // NOI18N
iiAlt.setName("iiAlt"); // NOI18N
iiUst.setText(resourceMap.getString("iiUst.text")); // NOI18N
iiUst.setName("iiUst"); // NOI18N
aaUst.setText(resourceMap.getString("aaUst.text")); // NOI18N
aaUst.setName("aaUst"); // NOI18N
aaAlt.setText(resourceMap.getString("aaAlt.text")); // NOI18N
aaAlt.setName("aaAlt"); // NOI18N
hUst.setText(resourceMap.getString("hUst.text")); // NOI18N
hUst.setName("hUst"); // NOI18N
hAlt.setText(resourceMap.getString("hAlt.text")); // NOI18N
hAlt.setName("hAlt"); // NOI18N
jLabel6.setText(resourceMap.getString("jLabel6.text")); // NOI18N
jLabel6.setName("jLabel6"); // NOI18N
jLabel7.setText(resourceMap.getString("jLabel7.text")); // NOI18N
jLabel7.setName("jLabel7"); // NOI18N
turKutu.setText(resourceMap.getString("turKutu.text")); // NOI18N
turKutu.setName("turKutu"); // NOI18N
jScrollPane1.setName("jScrollPane1"); // NOI18N
NOI18N
jTextArea1.setName("jTextArea1"); // NOI18N
jScrollPane1.setViewportViewView(jTextArea1);
jScrollPane2.setName("jScrollPane2"); // NOI18N
jPanel1.setName("jPanel1"); // NOI18N
graf4.setText(resourceMap.getString("graf4.text")); // NOI18N
graf4.setName("graf4"); // NOI18N
graf3.setText(resourceMap.getString("graf3.text")); // NOI18N
graf3.setName("graf3"); // NOI18N
graf2.setText(resourceMap.getString("graf2.text")); // NOI18N
graf2.setName("graf2"); // NOI18N
graf1.setText(resourceMap.getString("graf1.text")); // NOI18N
graf1.setName("graf1"); // NOI18N
graf5.setText(resourceMap.getString("graf5.text")); // NOI18N
graf5.setName("graf5"); // NOI18N
graf6.setText(resourceMap.getString("graf6.text")); // NOI18N
graf6.setName("graf6"); // NOI18N
graf7.setText(resourceMap.getString("graf7.text")); // NOI18N
graf7.setName("graf7"); // NOI18N
graf8.setText(resourceMap.getString("graf8.text")); // NOI18N
graf8.setName("graf8"); // NOI18N
javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    .addContainerGap(903, Short.MAX_VALUE)
);

```

```

        jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.BASELINE)
        .addComponent(graf5)
        .addComponent(graf4)
        .addComponent(graf3)
        .addComponent(graf2)
        .addComponent(graf1)
        .addComponent(graf6)
        .addComponent(graf7)
        .addComponent(graf8))
        .addContainerGap(409, Short.MAX_VALUE))
);
jScrollPane2.setViewportView(jPanel1);
jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
jLabel1.setName("jLabel1"); // NOI18N
jLabel8.setFont(resourceMap.getFont("jLabel8.font")); // NOI18N
jLabel8.setText(resourceMap.getString("jLabel8.text")); // NOI18N
jLabel8.setName("jLabel8"); // NOI18N
jLabel9.setText(resourceMap.getString("jLabel9.text")); // NOI18N
jLabel9.setName("jLabel9"); // NOI18N
hacUst.setText(resourceMap.getString("hacUst.text")); // NOI18N
hacUst.setName("hacUst"); // NOI18N
hacAlt.setText(resourceMap.getString("hacAlt.text")); // NOI18N
hacAlt.setName("hacAlt"); // NOI18N
javax.swing.GroupLayout mainPanelLayout = new
javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup())
        .addContainerGap()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup())

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup())

```

```

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addComponent(jLabel4)
    .addComponent(jLabel5))

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(65, 65, 65)
        .addComponent(jLabel6)
        .addGap(68, 68, 68)
        .addComponent(jLabel2)
        .addGap(74, 74, 74)
        .addComponent(jLabel3))
    .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(34, 34, 34)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
    .addComponent(iiUst)
    .addComponent(iiAlt,
javax.swing.GroupLayout.DEFAULT_SIZE, 61, Short.MAX_VALUE))
    .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

.addGroup(mainPanelLayout.createSequentialGroup())
.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
    .addComponent(aaUst)
    .addComponent(aaAlt,
javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
    .addComponent(hUst)
    .addComponent(hAlt,
javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(mainPanelLayout.createSequentialGroup())
    .addGap(39, 39, 39)
    .addComponent(jButton1))))))
    .addComponent(jLabel7))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
, 21, Short.MAX_VALUE)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)
    .addGroup(mainPanelLayout.createSequentialGroup())
        .addComponent(jLabel8)
        .addGap(348, 348, 348))
    .addGroup(mainPanelLayout.createSequentialGroup())
.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.TRAILING)
.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
    .addComponent(hacUst)
    .addComponent(hacAlt,
javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jLabel9,
javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 595,
javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 719,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap())
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 989, Short.MAX_VALUE)
);
mainPanelLayout.setVerticalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
    .addGroup(mainPanelLayout.createSequentialGroup())
    .addContainerGap()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup())

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
            .addComponent(jLabel3)
            .addComponent(jLabel9))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
            .addComponent(jLabel6)
            .addComponent(jLabel2)))
        .addGap(21, 21, 21)
        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.CENTER)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(iiAlt,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
            .addComponent(iiUst,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(aaAlt,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
            .addComponent(aaUst,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(hAlt,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
            .addComponent(hUst,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addComponent(jLabel4)
            .addGap(18, 18, 18)
            .addComponent(jLabel5)))
        .addGap(26, 26, 26)
    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
        .addComponent(jLabel7)
        .addComponent(turKutu,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton1)))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(35, 35, 35)
                .addComponent(hacAlt,
)
                    .addComponent(hacUst,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(26, 26, 26)
                .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
                .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 421, Short.MAX_VALUE))
    );

    menuBar.setName("menuBar"); // NOI18N

    fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
    fileMenu.setName("fileMenu"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(genalgoritma.GenAlgoritma
App.class).getContext().getActionMap(GenAlgoritmaView.class, this);

```

```

        exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
        exitMenuItem.setText(resourceMap.getString("exitMenuItem.text")); //
NOI18N
exitMenuItem.setToolTipText(resourceMap.getString("exitMenuItem.toolTipT
ext")); // NOI18N
        exitMenuItem.setName("exitMenuItem"); // NOI18N
        fileMenu.add(exitMenuItem);
        setComponent(mainPanel);
        setMenuBar(menuBar);
    }// </editor-fold>

```

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    iUst = Double.parseDouble(iiUst.getText());
    iAlt = Double.parseDouble(iiAlt.getText());
    aUst = Double.parseDouble(aaUst.getText());
    aAlt = Double.parseDouble(aaAlt.getText());
    lUst = Double.parseDouble(hUst.getText());
    lAlt = Double.parseDouble(hAlt.getText());
    turSayisi = Integer.parseInt(turKutu.getText());
    vUst = Integer.parseInt(hacUst.getText());
    vAlt = Integer.parseInt(hacAlt.getText());
    //ilk populasyon oluřturuluyor
    populasyonOlustur(300);
    QCseries = new XYSeries("QC - Nesil");
    lseries = new XYSeries("l - Nesil");
    Aseries = new XYSeries("A - Nesil");
    Hseries = new XYSeries("H - Nesil");
    Nseries = new XYSeries("N - Nesil");
    QCllseries = new XYSeries("QC - l");
    QCAseries = new XYSeries("QC - A");
    QCllseries = new XYSeries("QC - H");
    populasyonuBastir();
    for (int tur = 1; tur <= turSayisi; tur++) {
        konsolaBas("-----" + tur + ". tur-----");
        konsolaBas("En buyuk 16 kromozom yeni populasyona aktarildi.");
        enBuyukleriAl(16);
        caprazla(300);
        konsolaBas("Populasyon caprazlandi.");
        sirala();
        konsolaBas("Her bir birey, %1 olasılıkla mutasyona uęruyor.");
        mutasyon(100);
        sirala();
        for (int u = 0; u < 300; u++) {
            son();
        }
        konsolaBas("1. bireyden itibaren QC ler toplandi, toplami rastgele
cekilmis olan iki sayiya denk gelenler caprazlandi.");
    }
}

```

```

        sirala();
        populasyonuBastir();
        for (int ba = 0; ba < 50; ba++) {
            Kromozom graftemp = populasyon.elementAt(ba);
            QCseries.add(tur, graftemp.QC);
            lseries.add(tur, graftemp.i);
            Aseries.add(tur, graftemp.a * 1000000);
            Hseries.add(tur, graftemp.l * 1000);
            Nseries.add(tur, graftemp.n);
        }
    }
    jTextArea1.setText(sb.toString());
    jLabel1.setText("Bulunan en iyi termoelektrik modül: " +
populasyon.elementAt(0));
    sb = new StringBuilder();
    double tempII=populasyon.elementAt(0).l*1000;
    double tempII=populasyon.elementAt(0).i;
    double tempAA=populasyon.elementAt(0).a*1000000;
    for (double aaa = aAlt; aaa < aUst; aaa = aaa + 0.01) {
        Kromozom QCAtemp = new Kromozom(tempII, aaa, tempII, (int)
(vUst/(aaa*tempII)));

        BufferedImage image = chart.createBufferedImage(400, 400);
        BufferedImage imageI = chartI.createBufferedImage(400, 400);
        BufferedImage imageA = chartA.createBufferedImage(400, 400);
        BufferedImage imageH = chartH.createBufferedImage(400, 400);
        BufferedImage imageN = chartN.createBufferedImage(400, 400);
        BufferedImage imageQCI = chartQCI.createBufferedImage(400, 400);
        BufferedImage imageQCA = chartQCA.createBufferedImage(400, 400);
        BufferedImage imageQCL = chartQCL.createBufferedImage(400, 400);
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField aaAlt;
    private javax.swing.JTextField aaUst;
    private javax.swing.JPanel mainPanel;
    private javax.swing.JMenuBar menuBar;
    private javax.swing.JTextField turKutu;
    private final Timer messageTimer;
    private final Timer busylconTimer;
    private final Icon idleIcon;
    private final Icon[] busylcons = new Icon[15];
    private int busylconIndex = 0;
    private JDialog aboutBox;
    static Vector<Kromozom> populasyon = new Vector<Kromozom>();
    static Vector<Kromozom> yeniPopulasyon = new Vector<Kromozom>();
    static Random generator = new Random();
    //Sınırlar

```



```

public static void populasyonuBastir() {
    maxIndex = i;
}
}
konsolaBas("En buyuk kromozom:");
konsolaBas(populasyon.elementAt(maxIndex).toString());
}
public static void populasyonOlustur(int buyukluk) {
    populasyon.removeAllElements();
    while (populasyon.size() < buyukluk) {
        double i, a, l;
        int n;
        i = generator.nextDouble() * (iUst - iAlt) + iAlt;
        a = generator.nextDouble() * (aUst - aAlt) + aAlt;
        l = generator.nextDouble() * (lUst - lAlt) + lAlt;
        //n=generator.nextInt(nUst)+16;
        n = (int) (vUst / (a * l));
        Kromozom temp = new Kromozom(i, a, l, n);
        if (temp.COP < 0 || temp.QC < 0) {
            continue;
        }
        populasyon.add(temp);
    }
}
@SuppressWarnings("unchecked")
public static void enBuyukleriAl(int tane) {
    Vector<Kromozom> temp = new Vector<Kromozom>();
}
    yeniPopulasyon.add(temp.remove(maxIndex));
    maxQC = 0;
    maxIndex = 0;
}
//populasyon=yeniPopulasyon;
}
@SuppressWarnings("unchecked")
    int p = generator.nextInt(populasyon.size());
    int q = p;
    while (q == p) {
        q = generator.nextInt(populasyon.size());
    }
    Kromozom a = populasyon.elementAt(q);
    Kromozom b = populasyon.elementAt(p);
    yeniPopulasyon.add(yenia);
    yeniPopulasyon.add(yenib);
}
}

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : SALİMİ ALTAN, Mitra
 Uyuşu : T.C.
 Doğum tarihi ve yeri : 14.02.1966 İran
 Medeni hali : Evli
 Telefon : 0 (312) 442 39 56
 e-mail : salimi_mitra@yahoo.co.uk

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /Elk.-Bil. Eğitimi	2001
Öğretmenlik Sertifikası	Ankara Üniversitesi/ Eğit.Bilimler Fak.	1997
Lisans	Ankara Üniversitesi/ Mat. Bölümü	1991
Lise	motteheden Lisesi	1984

İş Deneyimi

Yıl	Yer	Görev
1998-2006	Başkent Üniversitesi	Öğretim Görevlisi
1997-1998	Bilay Dersanesi	Matematik Öğretmeni
1993-1996	Yumak Bilgisayar Kursu	Bilgisayar Öğretmeni

Yabancı Dil

İngilizce, Farsca

Yayınlar

1. Salimi Altan, M., Ahıska, R., “Termoelektrik Modüllerin Seçimi İçin Bilgisayar Programı”, **Bilgi teknolojileri kongresi II**, 141-143, 01-04 Mayıs 2003, Denizli
2. Salimi Altan, M., “Yapay Sinir Ağları İle Oyun Tasarımı”, **Bilgi teknolojileri kongresi III**, 190-192, 7-9 Ekim 2004, Denizli
3. Salimi Altan, M., Ahıska, R., “Isıl İşlemlerinde, Uzman Sistemleri” **Gap V. Mühendislik kongresi**, 614-620, 26-28 Nisan 2006, Urfa

Hobiler

Tenis, Resim, Doğa gezileri