

Enerji Sistemlerinde Yapay Arı Kolonisi (YAK) Algoritması Kullanarak Yük Akışı Optimizasyonu

Nihat Pamuk

Türkiye Elektrik İletim Anonim Şirketi (TEİAŞ), 5. İletim Tesis ve İşletme Grup Müdürlüğü, Sakarya

nihatpamuk@gmail.com.tr

Özet: Enerji sistemlerinin güvenli bir şekilde işletilmesi için, sisteme yeni üretim ve tüketim tesislerinin ilave edilmesi gerekmektedir. İlave edilecek iletim hatlarının devreye girmeden önce oluşturacağı etkilerinin bilinmesi gerekmektedir. Bu nedenle enerji iletim hatlarındaki maksimum yüklenme kapasite sınır değerleri aşılmadan yük akışlarının yapılması gerekmektedir. Son zamanlarda enerji sistemlerinde yük akışı problemlerinin çözümünde birçok sezgisel yöntem kullanılmaktadır. Bu sezgisel yöntemler arasında en doğru sonuç verenlerden birisi yapay arı kolonisi (YAK) algoritmasıdır. YAK algoritması doğada yiyecek arayan arılardan esinlenerek geliştirilmiştir. Bu algoritma küresel ve yerel uzayı komşuluk prensibine göre çalışmaktadır. Bu çalışmada, yük akışı eşitlikleri altında iletim hattı kayıplarını minimum yapan kontrol değişkenleri YAK algoritması yardımıyla PowerWorld benzetim programı kullanılarak hesaplanmıştır.

Anahtar Sözcükler: Enerji Sistemleri, Yük Akışı, Optimizasyon, Yapay Arı Kolonisi.

The Optimization of Load Flow in Energy Systems Using Artificial Bee Colony

Abstract: Energy systems need to be added to new production and consumption plant for safe operation. The effects of transmissions lines must be known before commissioning. Therefore, load flow must be done without exceeding maximum load capacity limit values of power transmission lines. In these days, many heuristic methods are used for load flow problems in power systems. Artificial bee colony (ABC) algorithm is one of the most accurate methods of these heuristic methods. ABC algorithm was developed by bees looking for food in nature. This algorithm works on the principle of global and local space neighborhood. In this study, control variables which minimize the loss of the transmission line were calculated under the load flow equations by the ABC algorithm using PowerWorld simulation program.

Keywords: Energy Systems, Load Flow, Optimization, Artificial Bee Colony.

1. Giriş

Güç sistemlerinde kullanılan ekipmanların fiziksel sınırlarını ve işletme limitlerini aşmadan, sistemdeki jeneratörlere üretimin paylaştırılması ve baralar arasındaki en uygun güç alış verişinin sağlanması optimal yük akışı olarak tanımlanır [1]. Son yıllarda enerji iletim sistemleri üzerinde uğraşılan problemlerden birisi optimal yük akışı problemi. Problemin amacı eşitlik ve eşitsizlik kısıtlamalarını sağlayarak, enerji

sisteminin enerji üretim maliyetinin minimizasyonudur. Enerji sistemlerindeki kontrol değişkenleri, salınım barası hariç diğer jeneratör baralarının aktif çıkış güçleri, jeneratör baralarının gerilim genlik değerleri, transformatör kademe değerleri ve şönt kapasite değerleridir. Bu problemin çözümü için literatürde birçok sayısal metot kullanılmıştır. Bu metotlara örnek olarak doğrusal olmayan programlama [2], kvadratik programlama [3], lineer programlama [4] ve newton tabanlı teknikler [5] gösterilebilir.

Son yıllarda gelişen sezgisel metotlar optimal yük akışı problemine de başarıyla uygulanmaktadır. Bu sezgisel metotlara örnek olarak genetik algoritma [6], parçacık sürü algoritması [7] ve evrimsel programlama [8] gösterilebilir.

2. Optimal Yük Akışı

Optimal yük akışı, aşağıdaki gibi tanımlanır; $f(x,u)$ fonksiyonunu $g(x,u)=0$ ve $h(x,u)\leq 0$ kısıtlamaları altında minimize etmektir. Burada $f(x,u)$ minimum değeri bulunmak istenen amaç fonksiyonudur. $g(x,u)$ yük akışı eşitliklerini göstermekte olup, $h(x,u)$ ise güvenlik limit değerlerini temsil etmektedir [18]. x ve u sırasıyla durum ve kontrol değişkenlerini göstermektedir. Durum değişkenleri salınım barasının aktif çıkış gücü $P_{salınım}$, yük baralarının gerilim genlik değerleri V_L ve jeneratör baralarının reaktif çıkış güçleri Q_g 'dir [9].

$$x^T = [P_{salınım}, V_L, Q_g] \quad (1)$$

Kontrol değişkenleri ise, salınım barası haricindeki jeneratör baralarının aktif çıkış gücü P_g , jeneratör baralarının gerilim genlik değerleri V_g , transformatörlerin kademe ayar değerleri T ve şönt kapasitelerin değerleri Q_c 'dir.

$$u^T = [P_g, V_g, T, Q_c] \quad (2)$$

Tüm enerji sisteminin toplam enerji üretim maliyeti (3) denklemi ile hesaplanmaktadır [9].

$$F \cos t = \sum_{i=1}^{N_g} (a_i + b_i P_{gi} + c_i P_{gi}^2) \quad (3)$$

Burada N_g sistemdeki toplam jeneratör sayısını, P_{gi} i . barada üretilen aktif güçleri, a_i , b_i , ve c_i , jeneratör yakıt maliyeti katsayılarını göstermektedir. Optimal yük akışı eşitlikleri, jeneratör aktif güç, jeneratör reaktif güç, bara

gerilim genlik, transformatör kademe değeri ve şönt kapasite limit değerleri (4) – (10) denklemleri ile gösterilmiştir [9].

$$i = 1, \dots, N;$$

$$P_{gi} - P_{li} - V_i \sum_{j=1}^N V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0 \quad (4)$$

$$Q_{gi} + Q_{ci} - Q_{li} - V_i \sum_{j=1}^N V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) = 0 \quad (5)$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max} \quad i = 1, \dots, N_g \quad (6)$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max} \quad i = 1, \dots, N_g \quad (7)$$

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad i = 1, \dots, N \quad (8)$$

$$T_i^{\min} \leq T_i \leq T_i^{\max} \quad i = 1, \dots, N_T \quad (9)$$

$$Q_{ci}^{\min} \leq Q_{ci} \leq Q_{ci}^{\max} \quad i = 1, \dots, N_c \quad (10)$$

P_{gi} ve Q_{gi} i . jeneratörün aktif ve reaktif çıkış güçlerini, P_{li} ve Q_{li} i . baradaki aktif ve reaktif yük değerlerini, Q_{ci} i . baradaki şönt kapasite değerini, N , N_g , N_T ve N_c sırasıyla sistemdeki toplam bara, jeneratör, transformatör ve şönt kapasite sayılarını göstermektedir. P_{gi}^{\min} , P_{gi}^{\max} , Q_{gi}^{\min} , Q_{gi}^{\max} , V_i^{\min} , V_i^{\max} , T_i^{\min} , T_i^{\max} , Q_{ci}^{\min} , Q_{ci}^{\max} ise ilgili değişkenlerin minimum ve maksimum değerlerini göstermektedir. Amaç fonksiyonunun değeri yük akışı ve sınırlama eşitlikleri altında (11) denklemi ile hesaplanmaktadır [9].

$$f_i(x_i, u_i) = F \cos t + R_1 |P_{salınım} - P_{salınım}^{\lim}| + R_2 \sum_{i=1}^{N_{PQ}} |V_i - V_i^{\lim}| + R_3 \sum_{i=1}^{N_g} |Q_{gi} - Q_{gi}^{\lim}| \quad (11)$$

Bu ifadede R_1 , R_2 ve R_3 büyük pozitif penaltı değerleridir. Durum değişkenlerinin limit değerleri $P_{salınım}^{\lim}$, V_i^{\lim} ve Q_{gi}^{\lim} 'dir. Bu değerler (12) - (14) denklemleri ile hesaplanmaktadır [9].

$$P_{salınım}^{\lim} = \begin{cases} P_{salınım}^{\max}; & P_{salınım} > P_{salınım}^{\max} \\ P_{salınım}^{\min}; & P_{salınım} < P_{salınım}^{\min} \end{cases} \quad (12)$$

$$V_i^{\text{lim}} = \begin{cases} V_i^{\text{max}} ; V_i > V_i^{\text{max}} \\ V_i^{\text{min}} ; V_i < V_i^{\text{min}} \end{cases} \quad (13)$$

$$Q_{gi}^{\text{lim}} = \begin{cases} Q_{gi}^{\text{max}} ; Q_{gi} > Q_{gi}^{\text{max}} \\ Q_{gi}^{\text{min}} ; Q_{gi} < Q_{gi}^{\text{min}} \end{cases} \quad (14)$$

3. Yapay Arı Koloni (YAK) Algoritması

YAK ilk olarak 2005 yılında Derviş Karaboğa tarafından tasarlanmıştır. Gerçek parametre optimizasyonu için önerilen YAK, arı kolonilerinin davranışlarını temel alan bir optimizasyon algoritmasıdır [10–11]. YAK algoritması bilgisayar, endüstri ve hidrolik mühendisliğinden havacılık ve uzay bilimi gibi farklı bilim dallarında başarıyla uygulanmıştır [11–12]. Doğal yaşamda bal toplayan bir arı kolonisi içinde görev paylaşımı vardır. Kolonide arılar üç gruba ayrılır [13]. Bunlar;

- İşçi Arılar: İşçi arılar komşuluk prensibine dayanarak daha fazla nektarın olduğu besin kaynaklarını araştırırlar. Her bir besin kaynağında bir işçi vardır. Dolayısıyla işçi arı sayısı besin kaynağı sayısına eşittir [14].
- Gözcü Arılar: Gözcü arılar kovanda bekler ve diğer arıların besin kaynakları ile ilgili bilgileri dansla paylaştıktan sonra nektarın fazla olduğu besin kaynağına yönelirler.
- Kâşif Arılar: Yiyecek arama sürecinin başlangıcında kâşif arılar rastgele dağılırarak yiyecek aramaya başlarlar.

Algoritmada yiyecek kaynakları, optimize edilmeye çalışılan problemin olası çözümlerine karşılık gelmektedir. Bir kaynağa ait nektar miktarı, o kaynakla ifade edilen çözümün kalite değerini ifade etmektedir [15].

Algoritma işleyişi beş temel adımda gerçekleşmektedir [14–15].

- Bal kaynağı bölgelerinin ilk değerlerinin belirlenmesi,
- İşçi arıların belirlenen bal kaynaklarına yönlendirilmesi,
- Bal kaynağı olmaya aday bölgeler için olasılık hesaplamalarının yapılması,
- İşçi arıların aktardıklarına bağlı olarak, gözlemci arıların yeni bal kaynaklarını belirlemesi,
- Mevcut bal kaynaklarının kullanım dışı bırakılma kararının verilmesi şeklindedir.

Algoritmanın ilk adımında, bal kaynakları bölgelerine ait değerler (15) nolu denklem ile hesaplanmaktadır [14–15].

$$x_{ij} = x_j^{\text{min}} + \text{rand}(0,1) \cdot (x_j^{\text{max}} - x_j^{\text{min}}) \quad (15)$$

Denklemde j üretilen kaynak sayısını, i ise en uygun parametre sayısını temsil etmektedir. Algoritmanın ikinci adımında, her işçi arı toplam kaynak sayısının yarısına eşit sayıda yeni kaynak bulmaktadır. Yeni kaynak bulurken;

$$v_{ij} = x_{ij} + \varphi_{ij} (x_{ij} - x_{kj}) \quad (16)$$

(16) nolu denklemde $k = (\text{int}(\text{rand} \cdot S) + 1)$ dir ve $j=1, \dots, D$ dir. v_i vektörü üretildikten sonra, x_i vektöründeki çözümlerle karşılaştırılır ve iyi olan kaynağı kullanır. Üçüncü adımda gözcü arılar denklem (17)'deki olasılıkla bir besin kaynağı seçer.

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (17)$$

Kolonilerde rastgele araştırma yapan kâşif arılar bulunmaktadır. Bu arılar yiyecek ararken herhangi bir ön bilgi kullanmamakta, tamamen rastgele araştırma yapmaktadır [16]. Kâşif arılar işçi arıların arasından seçilmektedir. Bu seçim işlemi limit parametresine bağlı olarak yapılmaktadır. Bir kaynağı ifade eden çözüm belli sayıdaki

deneme ile geliştirilememişse bu kaynak terk edilir. Bu kaynağa gelip giden arıda kâşif arı olarak yeni nektar kaynağı aramaya gider. İşçi arının kaynağa gelip gitme sayısı limit parametresi ile belirlenir. Kâşif arının yeni bir kaynak bulması denklem (18)'den hesaplanmaktadır.

$$x_{ij} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) * \text{rand} \quad (18)$$

YAK algoritmasında işçi ve gözcü arılar keşfedilen kaynaktan faydalanma işleminde, kâşif arılar ise keşif işleminde görev alırlar. Arılar birim zamanda yuvaya getirilen yiyecek miktarını (E/T) belirten enerji fonksiyonunu maksimize etmek için çalışırlar. Bir en üst düzeye çıkarma probleminde de amaç fonksiyonunun $F(\theta_i)$, $\theta_i \in R^p$, en üst düzeye çıkarılması gerekmektedir. θ_i , i. kaynağın pozisyonu olmak üzere $F(\theta_i)$ bu nektar miktarına karşılık gelir ve $E(\theta_i)$ ile orantılıdır.

Gözcü arıların bir kaynağı seçmeleri $F(\theta)$ değerine bağlıdır. Kaynağın nektar miktarı ne kadar fazla olursa, bu kaynağı bir gözcü arı tarafından seçilme olasılığı o kadar fazla olmaktadır. Yani θ_i pozisyonundaki bir kaynağı seçme olasılığı denklem (19) 'dan hesaplanmaktadır.

$$P_i = \frac{F(\theta_i)}{\sum_{k=1}^S F(\theta_k)} \quad (19)$$

Gözcü arı, işçi arıların dansını izledikten ve (19) denklemindeki olasılık değeri ile θ_i konumundaki kaynağı seçtikten sonra, bu kaynağın komşuluğun da bir kaynak belirler ve kaynağın nektarını almaya başlar. Seçilen komşuya ait pozisyon bilgisi denklem (20)'den hesaplanmaktadır.

$$\theta_i(c+1) = \theta_i(c) \pm \phi(c) \quad (20)$$

$\theta(c)$, k ve i'den farklı rastgele üretilen nüfustaki bir çözüme ait indis olmak üzere

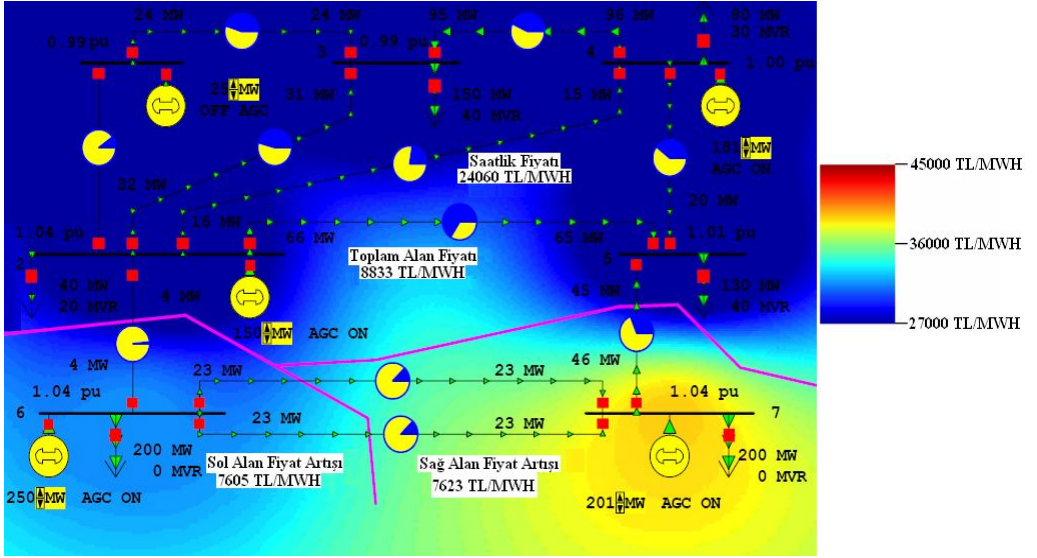
$\theta_i(c)$ ve $\theta_k(c)$ çözümlerinin bazı bölümlerinin farkının alınması ile hesaplanır. $\theta_i(c+1)$ 'e ait nektar miktarı $F(\theta_i(c+1))$, $\theta_i(c)$ konumundaki kaynağa ait nektar miktarından daha fazla ise arı kovana giderek bu bilgisini diğerleri ile paylaşır ve yeni pozisyon olarak $\theta_i(c+1)$ aklında tutar, aksi durumda $\theta_i(c)$ 'yi hafızasında saklamaya devam eder.

θ_i konumundaki nektar kaynağı "limit" parametresi sayısınca gelişmemiş ise θ_i 'deki kaynak terk edilir ve o kaynağın arısı kâşif arı haline gelir. Kâşif arı rastgele araştırma yapar ve yeni bir kaynak bulur. Bulunan yeni kaynak θ_i 'ye atanır. Algoritmadaki en iyi kaynağı bulma çabası her çevrimde devam eder. Algoritma maksimum çevrim sayısına ulaştığında sona erer. Elde edilen sonuçlar en uygun sonucu verir [16–17].

4. Örnek Güç Sistemi

Şekil 1'de 5 jeneratörlü 7 baralı örnek bir güç sisteminin tek hat şeması gösterilmiştir. Söz konusu 7 baralı güç sistemi daha rahat incelenebilmesi amacıyla üç bölgeye ayrılmıştır. PowerWorld benzetim programı yardımıyla bölgeler arasındaki yük akışları, fiyat artışları ve saatlik puant fiyatları YAK algoritması kullanılarak incelenmiştir.

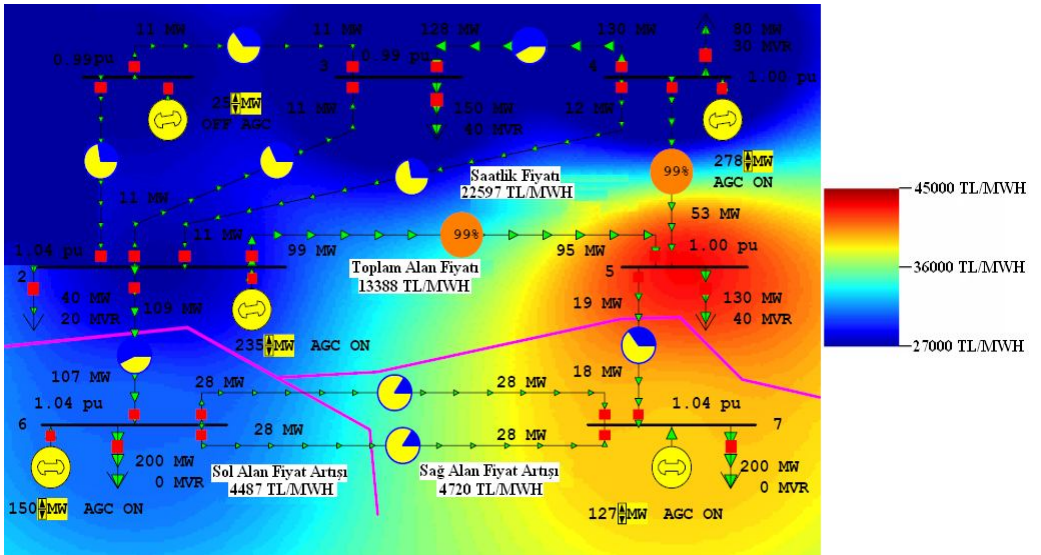
Şekil 1'deki sistemde, sağ bölgedeki enerji fiyat artışlarının yükseldiği görülmüştür. Enerjinin günlük puant fiyatı 24060 TL/MWH olarak belirlenmiştir. Söz konusu sisteme $P_{\text{yük}} = 767.9$ MW için, YAK algoritması uygulanarak optimal yük akışı yapılmıştır. Yapılan çalışmada ölçekleme faktörü $\gamma = 500$, yiyecek kaynağı sayısı (görevli arı sayısı) $SN = 20$, parametre sayısı $D = 5$, $TOL_{\text{hata}} = 1 \times 10^{-6}$ MW limit (çözüm geliştirememeye sayacı) 10 ve iterasyon sayısı 300 olarak alınmıştır. YAK algoritması güç sistemine uygulandığında ve sadece yakıt maliyeti göz önüne alındığında ($w=1$) elde edilen toplam yakıt maliyetinin 200 iterasyon sonucunda sabit bir değere ulaştığı görülmüştür.



Şekil 1. 7 Baralı Enerji Sistemindeki Yük Akışı

Şekil 2’de ise optimal yük akışı yapıldıktan sonraki sisteme ait tek hat şeması gösterilmiştir. Şekil 2 incelendiğinde sağ bölgeden üst bölgeye doğru olan yük akışında sınır değerlere ulaşıldığı fakat enerji birim fiyatında azalma olduğu tespit edilmiştir. Optimal yük akışı yapıldıktan sonraki

sistemin günlük puant fiyatı 22597 TL/MWH olarak belirlenmiştir. Enerjinin toplam bölge fiyatında ise bir artışın olduğu görülmüştür. Şekil 3 ‘de optimal yük akışı yapıldıktan sonraki PowerWorld benzetim programındaki sistemin kısıtlılık kayıtları gösterilmiştir.



Şekil 2. 7 Baralı Enerji Sistemindeki Optimal Yük Akışı

OPF Constraints Records

Line/Transformer Constraints

Disable Line/Transformer MVA Limit Enforcement

MVA Enforcement Percentage: 100.0

Percent Correction Tolerance: 2.0

Only Show Limit Violations

Only Show for Areas with Line MVA Enforcement

Interface Constraints

Disable Interface MW Limit Enforcement

MW Enforcement Percentage: 100.0

Percent Correction Tolerance: 2.0

Only Show Limit Violations

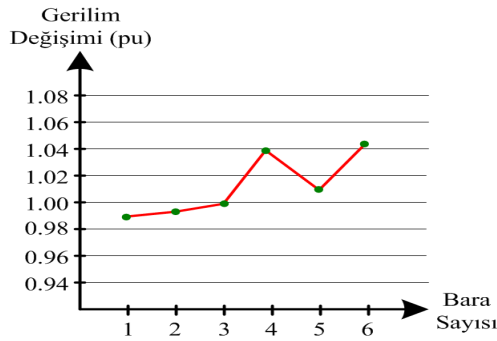
Only Show for Areas with Line MVA Enforcement

Lines/Transformers | Interfaces

	From Number	From Name	To Number	To Name	Circuit	Enforce MVA	Max MVA	Max Percent	Lim MVA	MVA Marg. Cost	MVA Unenforce	Constrai
1	1	1	2	2	1	YES	0.1	0.1	100.0			No
2	1	1	3	3	1	YES	100.1	100.1	100.0	12.0		Yes
3	2	2	3	3	1	YES	100.0	100.0	100.0	6.0		Yes

Şekil 3. 7 Baralı Enerji Sistemindeki Kısıtlılık Kayıtları

Şekil 4’de her bir bara gerilim değerinin (pu cinsinden) iterasyonlara göre değişimi gösterilmiştir.



Şekil 4. Baralarda Oluşan Gerilim Değişimleri (pu)

7 baralı enerji sistemindeki optimal yük akışı için elde edilen YAK algoritması sonuçları çizelge 1’de gösterilmiştir. Toplam yakıt maliyetinin yaklaşık olarak 180. iterasyondan sonra çok az değiştiği hatta 195. iterasyondan sonra neredeyse değişmediği görülmüştür. w ’nın değeri 0’ dan başlayarak 0,1 değerlerle 1’e doğru artırılırken elde edilen toplam yakıt maliyetinin azaldığı görülmektedir. İleriki çalışmalarda ($w=1$) kayıplı durumda

YAK algoritması sisteme uygulanırsa gerçek sistem değerlerine daha yakın sonuçlar elde edilecektir.

Çizelge 1. Güç Sistemine ait YAK Sonuçları

Çıkış Gücü Ağırlık (w)	Maliyet (TL/MWH)	Süre (sn)
1.0	8880.345	1.12575
0.9	8963.460	1.10567
0.8	8972.975	1.09368
0.7	9063.769	1.08642
0.6	9286.453	1.06798
0.5	9587.982	1.06346
0.4	9958.369	1.05698
0.3	10994.937	1.05357
0.2	11634.490	1.04424
0.1	13002.845	1.04507
0.0	13392.273	1.04147

5. Sonuçlar

YAK algoritması parametre sayısının az olması sebebiyle basit ve esnek bir algoritmadır. Araştırma uzayını gruplara ayırarak araştırma yapabilmesi, düşük popülasyon değerlerinde, yani daha az

çevrim zamanında optimum sonuca yakın bir yakıt maliyeti elde edilmesini sağlamaktadır. YAK algoritması durdurma kriteri sağlanana kadar birçok kez çalıştırılmış ve en iyi çözümü veren değerler sonuç olarak alınmıştır. Bu çalışmada durdurma kriteri olarak çevrim sayısı dikkate alınmıştır. Algoritmanın yüz defa çalıştırılması ile yaklaşık yüz ellinci çevrimde sonuca gidilebildiği görülmüş ve güvenilirliği sağlamak adına çevrim sayısı artırılmıştır. YAK algoritması diğer metotlara göre araştırma uzayını hızlı bir şekilde tarayabilmesi daha büyük güç sistemlerinde daha hızlı sonuçlar elde edilmesini sağlayacaktır.

6. Kaynaklar

- [1] Somasundaram, P., “Evolutionary Programming Based Security Constrained Optimal Power Flow”, *Electric Power System Research*, 72(1): 137–145 (2004).
- [2] Habiabollahzadeh, H., Luo, G.X., Semlyen, A., “Hydrothermal Optimal Power Flow Based on a Combined Linear and Nonlinear Programming Methodology”, *IEEE Transaction Power System*, 4(2): 530–537 (1989).
- [3] Lipowski, J.S., Charalambous, C., “Solution of Optimal Load Flow Problem by Modified Recursive Quadratic Programming Method”, *Proc. of IEE C128*, 4(1): 288–294 (1981).
- [4] Mota-Palomino, R., Quintana, V.H., “Sparse Reactive Power Scheduling by a Penalty-Function Linear Programming Technique”, *IEEE Transaction Power System*, 1(3): 31–39 (1986).
- [5] Zhang, S., Irving, M.R., “Enhanced Newton-Raphson Algorithm for Normal Controlled and Optimal Power Flow Solutions Using Column Exchange Techniques”, *IEE Proc. Generation Transmission Distribution*, 141(6): 647–657 (1994).
- [6] Iba, K., “Reactive Power Optimization by Genetic Algorithms”, *IEEE Transactions Power System*, 9(2): 685–692 (1994).
- [7] Chen, G., Yang, J., “A New Particle Swarm Optimization Solution to Optimal Reactive Power Flow Problems”, *Power and Energy Engineering Conference*, 1–4 (2009).
- [8] Wu, Q.H., Ma, J.T., “Power System Optimal Reactive Power Dispatch Using Evolutionary Programming”, *IEEE Transaction Power System*, 10(3): 1243–1249 (1995).
- [9] Arifoğlu, U., “Güç Sistemlerinin Bilgisayar Destekli Analizi”, *Alfa Yayınevi*, İstanbul, 265–304 (2002).
- [10] Karaboğa, D., Baştürk, B., “A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm”, *Journal of Global Optimization*, 39(3): 459–471 (2007).
- [11] Marinakis, Y., Marinaki, M., Matsatsinis, N., “A Hybrid Discrete Artificial Bee Colony-GRASP Algorithm for Clustering”, *Computers and Industrial Engineering*, Art. No. 5223810, 548–553 (2009).
- [12] Kang, F., Li, J.J., Xu, Q., “Hybrid Simplex Artificial Bee Colony Algorithm and its Application in Material Dynamic Parameter Back Analysis of Concrete Dams”, *Shuili Xuebao / Journal of Hydraulic Engineering*, 40(6): 736–742 (2009).
- [13] Akay, B., “Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi”, Doktora Tezi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri, (2009).

- [14] Karaboğa, D., “Yapay Zekâ Optimizasyon Algoritmaları”, *Atlas Yayın Dağıtım*, İstanbul, (2004).
- [15] Karaboğa, N., “A New Design Method Based on Artificial Bee Colony Algorithm for Digital IIR Filters”, *Journal of Franklin Institute*, 346(1): 328–348 (2009).
- [16] Bao, L., Zeng, J., “Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony Algorithm”, *9th International Conference on Hybrid Intelligent Systems*, (2009).
- [17] Singh, A., “An Artificial Bee Colony Algorithm for the Leaf-Constrained Minimum Spanning Tree Problem”, *Applied Soft Computing*, 9(1): 625–631 (2009).
- [18] Ayan, K., Kılıç, U., “Optimal Reaktif Güç Akışının Kaotik Yapay Arı Kolonisi ile Çözümü”, *6th International Advanced Technologies Symposium IATS’11 Fırat Üniversitesi*, 20–24, Elazığ, (2011).