

Lojistik-Singer Harita Tabanlı Yeni Bir Kaotik Sürü Optimizasyon Yöntemi

Fahrettin Burak DEMİR

Department of Computer Technologies, Doğanşehir Vahap
Küçük Vocational School, Malatya Turgut Özal University,
Malatya, Turkey
fahrettin.demir@ozal.edu.tr

Türker TUNCER

Department of Digital Forensics Engineering, Technology
Faculty, Firat University, Elazığ, Turkey
turkertuncer@firat.edu.tr

Adnan Fatih KOCAMAZ

Department of Computer Engineering,
Engineering Faculty, Inonu University, Malatya, Turkey
fatih.kocamaz@inonu.edu.tr

Özet— Günlük yaşamda pek çok problem, sonsuz çözüm uzayına sahip olduğu için klasik matematiksel yöntemler kullanılarak çözülememektedir. Bu nedenle, benzer problemlerin çözümünde, sonsuz çözüm uzayını küçülten ve matematiksel tahmin prensibine dayanan meta-sezgisel optimizasyon yöntemlerinin kullanılması önerilmektedir. Meta-sezgisel optimizasyon yöntemlerinin başarımını artırmak amacıyla sayı üretici ve parametre belirleyici olarak kaotik haritalar kullanılmaktadır. Bu makalede yeni bir kaotik optimizasyon yöntemi geliştirilmiş ve önerilen optimizasyon yönteminde lojistik ve singer harita kullanılmıştır. Önerilen yöntemin performansını test etmek amacıyla literatürde sıkça kullanılan 6 farklı kıyaslama fonksiyonu ve 3 farklı sürü tabanlı optimizasyon yöntemi kullanılmıştır. Önerilen yöntem bütün fonksiyonlar için daha optimum sonuçlar üretmiştir. Ve bu sayede sürü optimizasyon yöntemlerinin lokal çözümlere takılması önlenmeye çalışılmıştır.

Index Terms—Kaotik optimizasyon, lojistik harita, singer harita, sayısal fonksiyonlar, sürü optimizasyonu.

Abstract— Many problems in daily life cannot be solved by using classical mathematical methods for having an infinite solution space. Therefore, it is recommended to use meta-heuristic optimization methods that reduce the infinite solution space and based on the mathematical prediction principle in solving similar problems. In order to increase the performance of meta-heuristic optimization methods, number generator and chaotic maps are used. In this article, a new method of chaotic optimization has been developed and logistic and singer maps are used in the proposed optimization method. In order to test the performance of the proposed method, 6 different benchmarking functions and 3 different swarm-based optimization methods were used. The proposed method has produced more optimum results for all functions. In this way, it has been tried to prevent the integration of swarm optimization methods into local solutions.

Index Terms— Chaotic optimization, logistic map, singer map, numerical functions, swarm optimization.

I. GİRİŞ

Geleneksel matematik yöntemleriyle çözülmesi neredeyse imkansız olan problemleri çözmek için meta-sezgisel optimizasyon algoritmaları literatürde sıklıkla kullanılmaktadır

[1]. Optimizasyon yöntemlerinin temel amacı, çözüm uzayında küresel bir optimum nokta bulmaktır. Optimizasyon yöntemleri, problemleri çözmek için hedef fonksiyona sürekli rastgele değerler gönderir ve bu şekilde en iyi değerleri bellekte tutar [2,3].

Meta-sezgisel algoritmalar genel olarak, evrimsel algoritmaları, doğa temelli sistemleri ve matematiksel yapıları modellemektedir [3]. Bu modellemeler içerisinde sıklıkla kullanılan genetik algoritmalar, Darwin'in evrim teorisinden yola çıkarak, başlangıç popülasyonu oluşturma, mutasyon ve elitist seçim gibi modellemeler gerçekleştirmiştir [4]. Doğa temelli optimizasyon algoritmaları, kuş, böcek, ağaç, su gibi yapıları modellemektedir. Canlı sürülerini modelleyen en bilinen yöntemlerden birisi Parçacık Sürü Algoritması (PSO)'dır [5]. PSO, en iyi parçacık seçimi, hız parametresinin belirlenmesi, parçacık değerlerinin güncellenmesi ve en iyi çözüm adımlarını güncelleme gibi adımlardan meydana gelmektedir [6]. Matematiksel yapıları modelleyen algoritmalarından birisi olan Sinüs-Kosinüs Algoritması ise Sinüs ve Kosinüs fonksiyonlarını kullanarak en iyi çözüme gitmeyi amaçlayan bir algoritmadır [7,8].

Pek çok optimizasyon yöntemi rastsallık ilkesine göre hareket etmektedir. Ve üretilen rastsal değerleri iyileştirmek için literatürde sıklıkla kullanılan ve iyi istatistiksel özelliklere sahip olan haritalar kullanılmaktadır. Bu kaotik haritalar, sayı üretici ve hız parametresi olarak literatürdeki pek çok yöntemde kullanılmıştır [9-13].

Bu makalede lojistik ve singer haritaları kullanılarak yeni bir hibrit kaotik harita önerilmiştir. Önerilen bu yeni harita kullanılarak yeni bir kaotik optimizasyon yöntemi önerilmiştir. Bu çalışmanın teknik anlamda katkıları aşağıda verilmiştir.

- Önerilen yeni kaotik hibrit harita, lojistik ve singer haritaları kullanılarak oluşturulmuştur. Ve oluşturulan yeni hibrit harita daha geniş kaotik alana sahiptir.
- Önerilen kaotik sürü optimizasyon algoritmasında sayı üretici ve parçacık güncellemeleri için önerilen yeni lojistik-singer haritası kullanılmaktadır.
- Önerilen kaotik optimizasyon yöntemini test amacıyla literatürde sıklıkla kullanılan 6 farklı kıyaslama fonksiyonu kullanılmıştır. Bütün kıyaslama fonksiyonları için en optimum sonuçların bulunduğu gözlemlenmiştir.

II. YÖNTEM VE HAZIRLIK

Bu makalede yeni ve güncel bir kaotik harita elde edebilmek için lojistik ve singer harita kullanılmıştır. Elde edilen yeni kaotik harita ile yeni bir sürü tabanlı optimizasyon algoritması oluşturulmuş ve 6 farklı kıyaslama fonksiyonu kullanılarak, 3 farklı optimizasyon yöntemiyle kıyaslanmıştır. Bu kısımda lojistik ve singer harita anlatılmıştır.

A. Lojistik Harita

Lojistik harita, literatürde en sık kullanılan kaotik haritalardan birisidir. Bu harita literatürde rastsal sayı üretme, imge şifreleme, kriptoloji ve optimizasyon amacıyla kullanılmaktadır. Lojistik haritanın matematiksel tanımı Denklem(1)'de verilmiştir. Bu harita, 0,1 aralığında rastsal sayılar üretmektedir [14].

$$x_{i+1} = rx_i(1 - x_i), r \in (3.57, 4), x \in (0, 1) \text{ and } x \neq \{0.25, 0.5, 0.75\} \quad (1)$$

B. Singer Haritası

Singer haritası literatürde sıklıkla kullanılan kaotik haritalardan birisidir ve singer haritasının matematiksel tanımı Denklem(2)'de verilmiştir. Bu harita da, 0,1 aralığında rastsal sayılar üretmektedir [15].

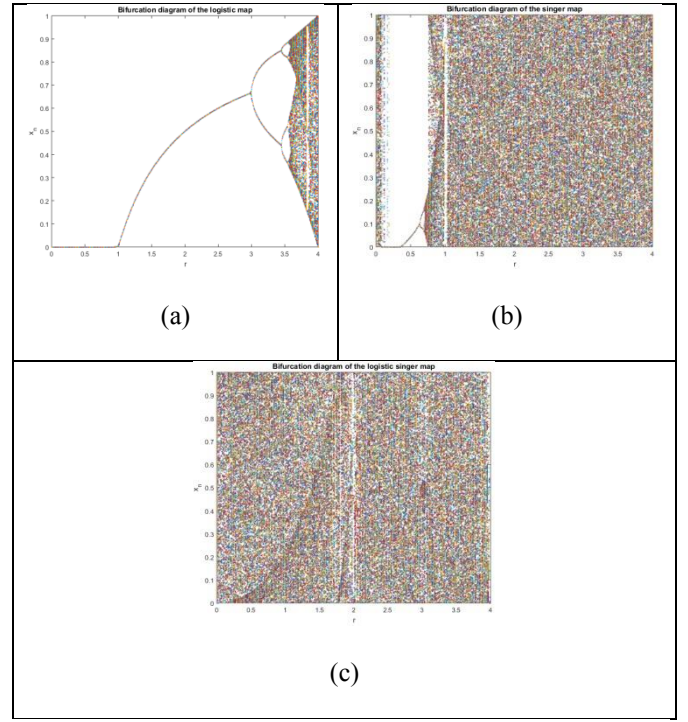
$$x_{i+1} = m(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4) \quad (2)$$

C. Hibrit Lojistik-Singer Harita

Bu makalede yeni bir hibrit kaotik harita önerilmiştir. Önerilen harita lojistik ve singer haritaları birlikte kullanılmaktadır. Bu haritanın temel amacı lojistik ve singer haritalarının güçlü yönlerini bir araya getirerek daha iyi bir hibrit kaotik harita oluşturmaktır. Önerilen lojistik-singer haritasının matematiksel tanımı Denklem(3)'te verilmiştir.

$$x_{n+1} = \text{mod} \left((4 - r) \left(7.86rx_n(1 - x_n) - 23.31((4 - r)x_n(1 - x_n))^2 + 28.75(rx_n(1 - x_n))^3 - 13.302875((4 - r)x_n(1 - x_n))^4 \right), 1 \right) \quad (3)$$

Kaotik haritalara ait çatallanma diyagramları(bifurcation diagrams) Şekil.1'de verilmiştir. Şekil.1. incelendiğinde, önerilen yeni lojistik-singer haritasının, lojistik ve singer haritalarından daha geniş bir kaotik aralığa sahip olduğu görülmektedir.



Şekil 1. Kaotik Haritalara ait çatallanma grafiği(bifurcation diagram) (a) lojistik harita (b) singer harita (c) lojistik-singer harita ($x \in (0, 1)$ ve $r \in (0, 4)$)

III. ÖNERİLEN LOJİSTİK-SINGER HARİTA TABANLI SÜRÜ OPTİMİZASYON YÖNTEMİ

Bu makalede lojistik-singer harita tabanlı yeni bir sürü optimizasyon yöntemi önerilmiştir. Bu yöntemde temel amaç aranan en optimum noktayı bulmaktır. Kaotik arama algoritmaları literatürde ilk defa Hamaizia ve ark. [16] tarafından önerilmiştir. Hamaizia ve ark. makalesinde çok boyutlu arama işlemi yaparken, bu makalede tek boyutlu kaotik harita kullanılmaktadır. Ayrıca adım büyüklüğü yerine lojistik-singer haritasının ürettiği değerler kullanılmaktadır. Aşağıda önerilen kaotik haritaya ait adımlar verilmiştir.

Adım 1: Başlangıç popülasyonunu rastsal olarak oluştur.

$$p_i = \text{rand}[0, 1] \times (UB - LB) + LB, i = \{1, 2, \dots, n\} \quad (4)$$

Yukarıdaki denklemde p parçacığı, rand[0,1] 0 ile 1 arasında rastgele değer üreten fonksiyon, UB arama uzayının üst sınırı, LB arama uzayının alt sınırı ve n kullanılacak parçacık sayısını ifade etmektedir.

Adım 2: En iyi değeri seç.

$$p_{best} = \min(f(p_i)) \quad (5)$$

f amaç fonksiyonunu ifade etmektedir.

Adım 3: Eşitlik 4'ü kullanarak tohum değerini oluştur.

$$x_1 = \frac{k}{iter + 1}, h = \frac{4k}{iter + 1} \quad (6)$$

k döngü değişkenin o anki değeri, iter maksimum döngü sayısı ve x_1 tohum değeri olarak ifade edilmektedir.

Adım 4: Her bir parçacık için Eşitlik 3'ü kullanarak x sayı dizisini oluştur.

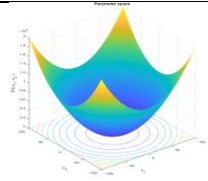
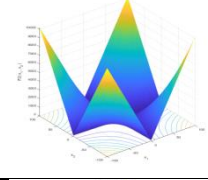
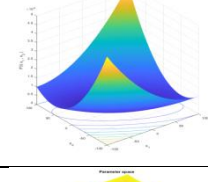
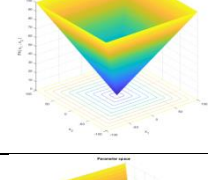
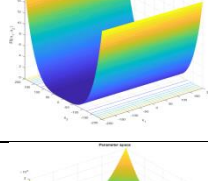
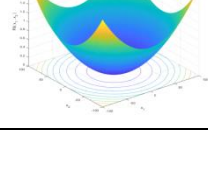
Adım 5: Parçacıkları güncelle.

$$p_i = p_i + x_i(p_{best} - p_i) \quad (7)$$

Adım 6: Eğer parçacık değeri sınır değerlerini aşmışsa parçacık aşağıdaki gibi güncellenir.

$$p_i = \begin{cases} p_{best}, p_i < LB \\ p_{best}, p_i > UB \end{cases} \quad (8)$$

Tablo.1. Kullanılan test setinin matematiksel ifadeleri ve grafiksel gösterimi [13,17].

Fonksiyon numarası	Fonksiyonun adı	Fonksiyonun matematiksel notasyonu	Fonksiyonun grafiksel gösterimi
F1	Sphere Fonksiyonu	$f_1(x) = \sum_{i=1}^m x_i^2, x \in [-100,100]^m$ (5)	
F2	Schwefel 2.22 fonksiyonu	$f_2(x) = \sum_{i=1}^m x_i + \prod_{i=1}^m x_i , x \in [-10,10]^m$ (6)	
F3	Schwefel 1.2. fonksiyonu	$f_3(x) = \sum_{i=1}^m \left(\sum_{j=1}^i x_j \right)^2, x \in [-100,100]^m$ (7)	
F4	Schwefel 2.21 fonksiyonu	$f_4(x) = \max\{ x_i , 1 \leq i \leq m\}, x \in [-100,100]^m$ (8)	
F5	Rosenbrock's fonksiyonu	$f_5(x) = \sum_{i=1}^{m-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], x \in [-30,30]^m$ (9)	
F6	Step fonksiyonu	$f_6(x) = \sum_{i=1}^m x_i + 0.5 ^2, x \in [-100,100]^m$ (10)	

Adım 7: En iyi parçacığı güncelle.

Adım 8: Global optimum noktasına veya maksimum iterasyon sayısına ulaşıncaya dek Adım 3-7' yi tekrarla.

IV. DENEYSEL SONUÇLAR VE TARTIŞMALAR

Bu kısımda önerilen yöntemin performansı 6 adet kıyaslama fonksiyonu kullanılarak test edilmiştir. Ayrıca bu fonksiyonlardan elde edilen değerler literatürde yaygın olarak kullanılan optimizasyon yöntemleriyle de karşılaştırılmıştır. Test seti olarak kullanılan fonksiyonlar Tablo.1.' de verilmiştir.

Elde edilen sonuçların anlamlı bir şekilde yorumlanabilmesi için önerilen Lojistik-Singer Harita Tabanlı Sürü Optimizasyonu Algoritması, Kaotik Bio-coğrafya Optimizasyonu(KBO) Algoritması [15], Kaotik Kril Sürüsü Optimizasyonu(KKSO) Algoritması [18] ve Kaotik Yerçekimi Arama Optimizasyonu(KYAO) Algoritması [19] yöntemleriyle karşılaştırılmıştır. Optimizasyon yöntemlerinin hepsi 40 popülasyon ve 500 maksimum iterasyon değerinde çalıştırılmıştır. Optimizasyon yöntemlerine ait parametreler Tablo.2.'de verilmiştir. Parametreleri belirlenen kaotik optimizasyon yöntemleri 30 kez çalıştırılarak en iyi değerler tespit edilmiş ve elde edilen ortalamalar ve standart sapmalar Tablo.3.'te gösterilmiştir.

Tablo.2. Optimizasyon yöntemlerinin kullandığı parametreler.

Metot	Parametreler
KBO	$\mu=0.005, \mu=0.8$
KKSO	$N^{\max}=0.01, V_f=0.02, D^{\max}=0.005$
KYAO	$G_0=100, \alpha=20$
LS-KSO	$x_1=k/\text{iter}, r=4x1$

Tablo.3. Kaotik haritalı yöntemlerle karşılaştırma sonuçları

Fonksiyon	Kriter	KBO	KKSO	KYAO	LS-KSO
F1	Ortalama	1.77×10^0	7.89×10^{-1}	1.18×10^{-16}	0
	S.S.	8.92×10^{-1}	2.15×10^{-1}	2.14×10^{-17}	0
F2	Ortalama	1.20×10^{-1}	2.31×10^2	2.70×10^{-1}	0
	S.S.	6.60×10^{-2}	3.03×10^1	5.69×10^{-1}	0
F3	Ortalama	7.55×10^3	4.31×10^2	6.55×10^2	0
	S.S.	3.23×10^3	1.31×10^2	2.69×10^2	0
F4	Ortalama	1.71×10^1	9.71×10^0	4.86×10^0	0
	S.S.	3.00×10^0	1.44×10^0	1.23×10^0	0
F5	Ortalama	4.18×10^2	1.91×10^2	5.41×10^1	0
	S.S.	2.96×10^2	1.24×10^2	5.16×10^1	0
F6	Ortalama	1.45×10^0	5.15×10^{-1}	7.77×10^{-17}	0
	S.S.	6.30×10^{-1}	1.90×10^{-1}	2.40×10^{-17}	0

Tablo.3.'te görüldüğü gibi, önerilen yöntem literatürde daha önceden önerilmiş ve yaygın olarak kullanılan 3 adet optimizasyon yöntemiyle 6 adet sayısal uygunluk fonksiyonu kullanılarak karşılaştırılmıştır. Karşılaştırma sonuçları önerilen yöntemin tüm fonksiyonlarında en iyi değeri elde ettiğini ve hatta bütün değerler için global optimum değerine ulaştığını açıkça göstermektedir. Veri setinde kullanılan tüm fonksiyonların optimum değerleri 0'dır ve şaşırtıcı olarak önerilen lojistik-singer harita tabanlı kaotik optimizasyon yöntemi tüm fonksiyonlarda optimum değerlere ulaşmıştır. Önerilen yöntemin sağladığı avantajlar aşağıdaki gibi sıralanmıştır.

- Bu makalede klasik kaotik haritalar kullanarak daha geniş bir kaotik aralığa sahip yeni bir hibrit kaotik harita tanımlanmıştır.
- Basit bir denklem ve önerilen kaotik harita bir arada kullanılarak yeni bir sürü tabanlı optimizasyon yöntemi önerilmiştir. Önerilen yöntem literatürde sıklıkla kullanılan 6 sayısal test fonksiyonuna uygulanmış ve optimum değerler elde edilmiştir.
- Önerilen kaotik optimizasyon tekniği basit bir matematiksel temele sahiptir. Bu sebepten dolayı bu yöntem gerçek dünya problemlerine kolayca uygulanabilir.
- Literatürde birçok optimizasyon yöntemi kaosu rastgele sayı üretmek için kullanılmaktadır. Bu makalede kaos doğrudan parçacıkların konumlarının belirlenmesi için

kullanılmıştır ve daha iyi sonuçlar elde edilmiştir. Bu yöntemle, lokal minimumda takılma problemi giderilmiştir. Önerilen makalenin dezavantajları ise;

- Küçük bir sayısal fonksiyon veri seti kullanması,
- Gerçek dünya problemlerine uygulanmamış olması olarak sıralanabilir.

V. SONUÇLAR VE ÖNERİLER

Sürü tabanlı optimizasyon yöntemlerini oluştururken yapılması gereken en önemli iş; parçacıkların nasıl güncelleneceğine karar vermektir. Bu çalışmada lojistik-singer haritası tabanlı yeni bir kaotik sürü optimizasyonu yöntemi önerilmiştir. Lojistik-singer haritasının çalışmaya en ciddi katkısı, rastsal sayı üretiminde iyi sonuçlar üretmesidir ve bu sebepten dolayı önerilen yöntemde parçacıklar lojistik-singer haritası kullanılarak güncellenmektedir. Önerilen yöntemi test etmek için literatürde sıklıkla kullanılmakta olan amaç fonksiyonları ve kaotik optimizasyon yöntemleri tercih edilmiştir. Sonuçlar incelendiğinde, önerilen yöntemin tüm test setinde başarılı sonuçlar ürettiği açık bir şekilde görülmektedir. Bu çalışma, ilerideki çalışmalarda optimizasyon yöntemlerinin bu yöntem ile çözülebileceğini ve kaotik haritaların hibrit bir şekilde uygulanabileceğini göstermektedir.

Gelecekteki çalışmalarda yeni nesil hibrit kaotik haritalar kullanılarak yeni kaotik optimizasyon teknikleri önerilebilir. Hibrit kaotik haritalar genellikle imge şifreleme yöntemlerinde kullanılmaktadır. Bu makaleyle birlikte hibrit kaotik haritaların optimizasyon alanında kullanımının artması beklenmektedir. Ayrıca bu tür yöntemlerin gerçek dünya problemlerini çözmek için kullanılacağı da düşünülmektedir.

KAYNAKLAR

- [1] Altay EV, Alatas B (2019). Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review*: 1-42.
- [2] Canayaz M, Karci A (2015). Investigation of cricket behaviours as evolutionary computation for system design optimization problems. *Measurement* 68:225-235
- [3] Mirjalili S., Lewis A. (2016). The Whale Optimization Algorithm, *Advances in Engineering Software*,95:51,67.
- [4] Zang W, Ren L, (2018). A cloud model based DNA genetic algorithm for numerical optimization problems, *Future Generation Computer Systems* 81 (2018) 465–477.
- [5] Ebenhart R, Kennedy J (1995). Particle swarm optimization, in: *Proceeding IEEE Inter Conference on Neural Networks*, 4, Perth, Australia, Piscataway, 1995, pp. 1942–1948.
- [6] Beheshti, Z, Shamsuddin SM, (2015). Non-parametric particle swarm optimization for global optimization, *Applied Soft Computing* 28 (2015) 345–359.
- [7] Mirjalili S (2016). SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowledge Based Systems* 96 (2016) 120–133. 7.
- [8] Tuncer, T. (2018). "LDW-SCSA: Logistic Dynamic Weight based Sine Cosine Search Algorithm for Numerical Functions Optimization." arXiv preprint arXiv:1809.03055.
- [9] Chen K., Zhou F., Liu A., Chaotic dynamic weight particle swarm optimization for numerical function optimization.
- [10] Zhou Y, Wu J, Ji L, Yu Z, Lin K, Hao L (2018). Transient stability preventive control of power systems using chaotic particle swarm optimization combined with two-stage support vector machine, *Electric Power Systems Research* 155 (2018) 111–120.
- [11] Mitic M, Vukovic N, Petrovic M, Miljkovic Z (2015). Chaotic fruit fly optimization algorithm, *Knowledge-Based Systems* 89 (2015) 446–458.
- [12] Naanaa A (2015) Fast chaotic optimization algorithm based on spatiotemporal maps for global optimization, *Applied Mathematics and Computation* 269 (2015) 402–411.
- [13] Demir FB, Tuncer T, Kocamaz AF (2019) "Lojistik-Gauss Harita Tabanlı Yeni Bir Kaotik Sürü Optimizasyon Yöntemi." *Anatolian Science-Bilgisayar Bilimleri Dergisi*: 47-53.
- [14] Zhenyu G, Bo C, Min Y, Binggang C (2006) Self-adaptive chaos differential evolution. In: Jiao L, Wang L, Gao X, Liu J, Wu F (eds) *Advances in natural computation*. Lecture notes in computer science, vol 4221. Springer, Berlin, Heidelberg, pp 972–975.
- [15] Saremi S, Mirjalili S, Lewis A (2014). Biogeography-based optimization with chaos, *Neural Comput & Applic* (2014) 25:1077–1097.
- [16] Hamaizia T, Lozi R, Hamri N (2012). Fast chaotic optimization algorithm based on locally averaged strategy and multifold chaotic attractor, *Applied Mathematics and Computation* 219 (2012) 188–196.
- [17] Yao X., Liu Y., Lin G., Evolutionary programming made faster, *IEEE Trans. Evol. Comput.* 3 (2) (1999) 82–102.
- [18] Wang GG, Guo L, Gandomi AH (2014). Chaotic krill herd algorithm, *Inf. Sci.* 274 (2014) 17–34.
- [19] Ju FY, Hong WC (2013) Application of seasonal SVR with chaotic gravitational search algorithm in electricity forecasting, *Appl. Math. Model.* 37 (23) (2013) 9643–9651