

**الگوریتم یادگیری بهینه سازی جستجوی عقبگرد و کاربردهای آن**

**چکیده**

الگوریتم جستجوی عقبگرد (BSA) پیشنهادی در این مقاله، یک الگوریتم تکاملی (EA) می باشد که برای حل مسائل بهینه سازی استفاده می شود. ساختار این الگوریتم ساده می باشد و تنها دارای یک پارامتر کنترلی است که باید تعیین شود. به منظور بهبود عملکرد همگرائی و گسترش دامنه استفاده از آن، یک الگوریتم جدید به نام یادگیری BSA (LBSA) در این مقاله ارائه شده است. در این روش، بطور سراسری از بهترین اطلاعات نسل حاضر و اطلاعات پیشین در BSA ، برای گزینش مجدد افراد باتوجه به احتمال تصادفی ترکیب می شوند و افراد باقی مانده موقعیت خود را توسط یادگیری دانش در بهترین فرد، بدترین فرد و یک فرد تصادفی از نسل حاضر دوباره انتخاب می کنند. دو مزیت مهم برای این الگوریتم وجود دارد. 1) بعضی از افراد موقعیت خود را با هدایت بهترین فرد ( آموزش دهنده ) بروزرسانی می کنند،که باعث همگرائی سریعتر می شود، 2) یادگیری از افراد متفاوت، بطور ویژه زمانی که بدترین فرد نادیده گرفته می شود، تنوع جمعیت را افزایش می دهد. برای آزمایش بهترین عملکرد LBSA، توابع سنجشی در CEC2005 و CEC2014 مورد آزمایش قرار گرفتند و این الگوریتم به منظور آموزش شبکه های عصبی برای پیش بینی سری های زمانی بی نظم و مشکلات مدلسازی سیستم های غیرخطی استفاده می شود. برای ارزیابی عملکرد BSA با بعضی از EAهای دیگر، چند مقایسه بین LBSA و دیگر الگوریتم های کلاسیک انجام شده است. نتایج نشان می دهند که LBSA با توجه به الگوریتم های دیگر عملکرد خوبی دارد و باعث بهبود عملکر الگوریتم BSA می شود.

**کلید واژه ها:** الگوریتم تکاملی. الگوریتم جستجوی عقبگرد. بهینه سازی مبتنی بر آموزش-یادگیری. آموزش شبکه های عصبی مصنوعی. مشکلات بهینه سازی

**1. مقدمه**

برخی از مشکلات علوم و مهندسی حل مسائل بهینه سازی می باشد، که امروزه محققان در تلاش برای جستجو و طراحی الگوریتم های بهینه سازی بهتر می باشند. برای حل مسائل پیچیده مانند توابع غیرخطی، مشتق ناپذیر و تابع هدف غیر- محدب، محققان بر طراحی الگوریتم های تکاملی (EA) جدید یا بهبود عملکرد الگوریتم های موجود متمرکز شده اند. از این الگوریتم ها، الگوریتم بهینه سازی هوش جمعی و الگوریتم های تکامل ژنتیکی در حل مسائل پیچیده بهینه سازی نقش بسیار مهمی دارند. الگوریتم بهینه سازی ازدحام ذرات (PSO) [12]، که به تقلید از جستجوی پرندگان می باشد، در استفاده برای بهینه سازی توابع [37] ، مسائل طبقه بندی شده ترتیبی [36] و سیستم های تولید برق [25] عملکرد خوبی دارد. برخی از انواع روش های PSO، مانند روش ازدحام ذرات کاملا آگاه (PSOFIPS) [20]، PSO مبتنی بر نسبت فاصله برازش (PSOFDR) [23] و یادگیری جامع ازدحام ذرات (CLPSO) [15]، می باشند که برای بهبود عملکرد الگوریتم PSO و حل مسائل بهینه سازی پیشنهاد شده اند. علاوه بر این، برخی از الگوریتم های ترکیبی نیز برای بهبود عملکرد الگوریتم PSO [13,21] معرفی شده اند و همچنین برخی روش های گسسته برای گسترش دامنه کاربرد الگوریتم PSO [2,8,10] پیشنهاد شده است. شبیه سازی فرآیند آموزش-یادگیری در الگوریتم بهینه سازی مبتنی بر یادگیری-آموزش (TLBO) [31] نیز ارائه شده است. در این روش پارامترهای کمتری وجود دارند که باید در معادلات بروزرسانی تعیین شوند، که این کار باعث می شود الگوریتم به آسانی اجرا شود. برای استفاده کامل از مزایای الگوریتم TLBO، دو روش در نظر گرفته شده است: یکی از این روش ها بهبود کارایی الگوریتم توسط تغییر فرآیند بروز رسانی یا ترکیب آن با دیگر EAها و از سوی دیگر گسترش دامنه های کاربردی آن است. طرح نخبه گرایی که در الگوریتم نخبه گرایی TLBO (ETLBO) استفاده شده است برای بهبود عملکرد خود برخی از بدترین یادگیرنده ها را در گروه هایی توسط نخبه های دیگر جایگزین می کند[29]. یک TLBO خود یادگیر که یک روش خود یادگیری جدید می باشد برای آموزش مدلسازی عصبی تابع پایه شعاعی (RBF) باتری ها استفاده شده است[41]. بسیاری از آموزش دهندها، از عامل آموزش تطبیقی، خودآموز آموزشی و یادگیری خود انگیزشی برای بهبود تنوع جمعیت استفاده می کنند[30]. الگوریتم اصلاح شده TLBO عملکرد خوبی برای انجام برخی از مسائل بهینه سازی داشته است. باتوجه به زمینه های کاربردی، عملگر جهش لِوی(Levy) [7] برای کمک به جلوگیری از بوجود آمدن نقطه بهینه محلی TLBO استفاده شده است و الگوریتم جدید با موفقیت برای سیستم IEEE سی شینه استفاده شده است. TLBO بهبود یافته[6] در جهتی یکسو برای افزایش توانایی جستجوی آموزش دهنده ها معرفی شده است که با یک الگوریتم تکاملی تفاضلی دوطرفه برای کنترل مسئله پخش بار بهینه توان راکتیو (ORPD) ترکیب شده است. روش های محلی و خود یادگیر [3] به منظور بهبود عملکرد TLBO ابتکاری استفاده شده است، این روش ها عملکرد خوبی از خود برای مسائل بهینه سازی سراسری نشان می دهند. بهینه سازی مبتنی بر آموزش-یادگیری چند هدفه[19] برای کنترل توان راکتیو سیستم ها استفاده می شود. بهینه سازی مبدل های حرارتی صفحه ای-پره دار[22] و فرآیندهای ماشینکاری مدرن [28]، مشکلات بهینه سازی برنامه ریزی واحدهای حرارتی آبی بلند مدت[32]، بهینه سازی سرد کن های حرارتی دو مرحله ای[27] و مشکلات برنامه ریزی دوباره خط تولید [11,14] توسط الگوریتم بهینه سازی TLBO حل شده اند. بررسی دقیق برنامه های کاربردی TBLO را می توان در مقاله [26] یافت. ساده سازی فرآیند آموزش و کاهش تعداد پارامترهای نامعین الگوریتم های بهینه سازی هوشمند موضوع بسیار مهمی می باشد. برای پرداختن به این موارد، یک الگوریتم جدید به نام الگوریتم جستجوی عقبگرد (BSA)[5] در سال 2013 پیشنهاد شده است که برای برخی از مشکلات بهینه سازی مهندسی استفاده می شود. الگوریتم BSA برای تجزیه و تحلیل موج سطحی [34] و برای حل مسائل بهینه سازی ترکیب آرایه های آنتن مدور متحدالمرکز استفاده می شود[9]. BSA و الگوریتم تکامل تفاضلی (DE) را می توان برای حل مسائل بهینه سازی بدون محدودیت باهم ترکیب نمود[40]. الگوریتم BSA مقابله ای [17] برای حل مسائل بهینه سازی، شناسایی پارامتر سیستم های فوق آشوبی معرفی شده است. الگوریتم BSA با سه روش کنترل محدودیت [43] برای حل مسائل بهینه سازی که دارای محدودیت می باشند استفاده می شود. برای برخی از EAهای کلاسیک، BSA الگوریتم بهینه سازی بسیار جدیدی می باشد و از طرفی نوع بهبود یافته الگوریتم BSA به نسبت کم تر مورد تحلیل قرار گرفته است.

همانطور که در بالا ذکر شده است، BSA و TLBO هر دو نشان دادند که در حل برخی از مسائل بهینه سازی بهتر از الگوریتم های دیگر می باشند، بویژه به دلیل آنکه الگوریتم ها ساده هستند و پارامترهای نامعین کمی در معادلات بروزرسانی فردی وجود دارد. از طرفی عملکرد الگوریتم های نوین در حل مسائل پیچیده نیاز به بهبود دارد. معایب این الگوریتم ها در بخش های 2 و 3 بصورت دقیق تر آمده است.

برای بهبود عملکرد سراسری الگوریتم BSA ، یک روش جدید که ترکیبی از ایده های TLBO و BSA می باشد در این مقاله ارائه شده است. بخش اصلی روش جدید به شرح زیر می باشد. بخش اول علاوه بر هدایت یادگیری در معادلات بروزرسانی BSA سرعت همگرایی BSA را بهبود می بخشد. بخش دوم برای ادغام سه روش یادگیری در یک معادله به منظور بروزرسانی افراد باتوجه به احتمالات تصادفی است. یکی دیگر از بخش ها اجتناب از استفاده بدترین فرد در جهت بهبود تنوع یا گوناگونی جمعیت می باشد. در مقابل برای دو تابع ارزیابی (FE) برای یک فرد در یک نسل از الگوریتم TLBO، تنها یک FE در هر نسل برای روش جدید وجود دارد. از این رو، هزینه محاسبات برای یک نسل از الگوریتم پیشنهادی نسبت به TLBO کمتر است.

بقیه مقاله به شرح زیر می باشد: BSA بنیادی و TLBO ابتکاری بطور مختصری به ترتیب در بخش 2 و بخش 3 معرفی شده اند؛ LBSA در بخش 4 اراده شده است؛ توابع سنجشی در CEC2005 و CEC2014 برای نشان دادن اثر الگوریتم های مختلف در بخش 5 آزمایش شده اند؛ برنامه های کاربردی برای دو مشکل مدلسازی غیرخطی و مشکلات پیش بینی سری های زمانی آشوبی در بخش 6 نمایش داده شده است و در نهایت نتیجه گیری در بخش 7 آمده است.

**2. BSA بنیادی**

BSA یک EA مبتنی بر جمعیت است که با استفاده از الگوریتم DE توسعه یافته است، اما شبیه به الگوریتم DE نمی باشد. این الگوریتم شامل 5 فرآیند می باشد: مقدار دهی اولیه، گزینش-1 ، جهش، ترکیب و گزینش-2. دو تنوع جمعیتی در BSA وجود دارد: یک از آنها جمعیت تکاملی و دیگری جمعیت آزمایشی می باشد. جمعیت آزمایشی از برخی اطلاعات پیشین جمعیت تکاملی تشکیل شده است و یک ماتریس جستجوی جهت دار توسط دو جمعیت برای بروزرسانی موقعیت افراد ایجاد می شود. تنها یک پارامتر کنترل، یعنی نرخی ترکیبی وجود دارد، که تعداد عناصر فردی را کنترل می کند و در یک آزمایش جهش می یابد. پنج مرحله از BSA به شرح زیر می باشد و جزئیات بیشتر آن در مقاله [5] بیان شده است.

1. مقداردهی اولیه. جمعیت اولیه و تاریخچه جمعیت BSA با توجه به معادله 1 و 2 مقدار دهی می شوند،که U توزیع یکنواخت می باشد و j پایین و j بالا مرزهای پایین و بالای متغییرها هستند.



که i فرد iام جمعیت است و i=1 , 2 , . . . , N، که N مقدار جمعیت می باشد، j بیت jام از فرد iام است، j=1 , 2 , . . . , D ، که D اندازه ابعاد متغییرها می باشد.

1. گزینش-1: در شروع هر تکرار ، تاریخچه جمعیت oldP در BSA باتوجه به رابطه 3 و 4 معرفی می شود:



1. جهش. فرم اولیه جمعیت آزمایشی BSA توسط عملیات جهش باتوجه به معادله (5) تولید می شود. همانطور که در معادله (5) نشان داده شده است، BSA با استفاده از مزیت تجربیات خود از نسل های قبلی به تولید یک جمعیت آزمایشی می پردازد؛ اطلاعات پیشین جمعیت ، کل فرآیند تکاملی را در برمی گیرد. F دامنه ماتریس جستجوی جهت دار را کنترل می کند. مقدار مشترک F برابر با 3•randn است که در آن randn ⁓ N(0 , 1)می باشد.



1. ترکیب . در BSA ، مقدار اولیه جمعیت آزمایشی از فرآیند جهش بوجود می آید و افراد آزمایشی با برازش بهتر برای تکامل افراد جامعه هدف استفاده می شوند. یک ماتریس با داده های مقداری صحیح باینری (نگاشت) با اندازه N • D ، مسیرهای ترکیب شده الگوریتم BSA را هدایت می کند. استراتژی ترکیب BSA در رابطه 6 بیان شده است.



گزینش-2 . در این مرحله، جمعیت نسل بعدی باتوجه به مکانیزم گزینش حریصانه تولید می شود. در روش پیشنهادی، برای مسائل کوچک، اگر برازش نسبت به کوچکتر باشد، جایگزین می شود. این فرآیند در رابطه 7 نشان داده شده است.



همانطور که در مقدمه ذکر شد، BSA الگوریتمی با ساختار ساده تر نسبت به برخی از EA ها، مانند PSOها، الگوریتم ژنتیگ و DEها می باشد. ترکیب و مکانیزم جهش از الگوریتم DE و انواع آن متفاوت می باشند. اطلاعات پیشین در فرآیند بروز رسانی برای بهبود عملکرد الگوریتم استفاده می شوند. با این حال، به عنوان یک الگوریتم نوین، BSA نیز معایبی دارد. اولین عیب آن نبودن هیچ هدایت کننده ای به عنوان بهترین فرد در فرآیند بروز رسانی می باشد که می تواند باعث سرعت همگرائی آهسته BSA شود. دومین عیب BSA، زمانی می باشد که تنوع جمعیت از بین رفته و افرد نمی توانند دوباره انتخاب شوند، زیرا جمعیت آزمایشی هنگامی که جمعیت تقریبا شبیه به نسل بعدی است ممکن نیست تغییر کند. این امر باعث کاهش توانایی BSA برای جلوگیری از ایجاد نقطه بهینه محلی می شود.

**3. مراحل اصلی TLBO**

TLBO نیز مانند EA مبتنی بر جمعیت می باشد که فلسفه آموزش و یادگیری در یک گروه را تقلید می کند. دو مرحله اصلی در TLBO وجود دارد: که مرحله آموزش دهنده و مرحله یادگیرنده می باشند. از آنجا که تنها ایده های یادگیری در الگوریتم پیشنهادی یادگیری BSA (LBSA) استفاده می شود، تنها دو مرحله اصلی الگوریتم TLBO در این بخش شرح داده شده است.

**3.1 مرحله آموزش دهنده**

در مرحله آموزش دهنده، آموزش دهنده دانش خود برای همه ی یادگیرنده ها در گروه توزیع می کند و یادگیرنده ها موقعیت خود را با توجه به آموزش دهنده و موقعیت میانگین گروه حاضر بروز رسانی می کنند. یادگیرنده با بهترین برازش برای آموزش دهنده شدن انتخاب می شود. برای یک مسئله بهینه سازی با ابعاد D، فرض کنید موقعیت یادگیرنده iام باشد، که در آن نشان دهنده راه حل میانگین گروه حاضر است و نشان هنده موقعیت آموزش دهنده می باشد. یادگیرنده iام موقعیت خود را باتوجه به تفاوت بین آموزش دهنده و موقعیت میانگین بروز رسانی می شود که به شرح زیر می باشد:



که و به ترتیب موقعیت های جدید و پیشین یادگیرنده iام می باشند و rand(•) عددی تصادفی در محدوده [0 , 1] می باشد. مکانیزم انتخاب حریصانه برای پذیرش بهتر و استفاده می شود. پس از آنکه موقعیت پذیرفته شود به مرحله یادگیرنده انتقال می یابد. ضریب آموزش ، مقدار میانگین را برای تغییر داده شدن تعیین می کند. مقدار بصورت ابتکاری بر روی 1 یا 2 تنظیم می شود. این تصمیم تصادفی با احتمال برابر به شرح زیر می باشد:



**3.2 مرحله یادگیرنده**

در هر تکرار، یادگیرنده k ام بصورت تصادفی به عنوان هدف یادگیری، یادگیرنده i ام انتخاب می شود. روش یادگیری، یادگیرنده i ام می تواند بصورت ریاضی به شرح زیر بیان شود:



که ، موقعیت جدید فرد i ام است و و ، به ترتیب موقعیت های پیشین i ام و k ام یادگیرنده ها هستند. علاوه بر این، اگر برازش نسبت به بهتر باشد پذیرفته می شود. ساختار TLBO ساده می باشد و از طرفی پارامترهایی که باید در معادلات بروز رسانی از پیش تعیین شده باشند در این الگوریتم وجود ندارد. به همین دلیل از این الگوریتم بطور گسترده برای حل مسائل بهینه سازی که داری قطعیت می باشند استفاده می شود. با این حال، این الگوریتم نیز دارای دو عیب می باشد. عیب اول آن است که دو برای هر فرد در هر نسل وجود دارد و از این رو هزینه محاسبه یک نسل نسبت به الگوریتمی با یک برای هر فرد در هر نسل بزرگتر است. عیب دوم آن است که دانش یادگیری از بهترین یادگیرنده ها و دو گزینش حریصانه از یک فرد ممکن است تنوع جمعیت را به سرعت کاهش دهد. به منظور بهبود عملکرد BSA، مکانیزمی برای یادگیری TLBO و یک عملگر مقابله کننده برای بهبود تنوع جمعیت در این مقاله پیشنهاد شده است. علاوه بر این، تنها یک از هر فرد در یک نسل وجود دارد، که در بخش بعدی توضیح داده می شود.

**4. LBSA**

**4.1 انگیزه**

انگیزه اصلی در ایجاد LBSA ، ترکیب ایده های یادگیری TLBO در BSA بنیادی برای بهبود عملکرد سراسری آن است. در الگوریتم BSA اولیه، زمانی که تنوع جمعیت در آنافاز ( مرحله‌ای از تقسیم میتوز ، که در آن دو نوار سازنده کروموزوم ، از هم جدا می‌شوند) تکاملی تضعیف می شود، توانایی تکامل افراد شروع به ضعیف شدن می کند. این پدیده فرد را از پیدا کردن بهینه سازی سراسری منصرف می کند. علاوه بر این، این عمل نشان می دهد که عملگر یادگیری می تواند سازگاری جمعیت را بهبود بخشد و یادگیری از بهترین فرد می تواند سرعت همگرایی الگوریتم را بهبود دهد و همچنین ظرفیت بهره برداری از الگوریتم را افزایش دهد. در هر حال، این پدیده تنوع جمعیت و ظرفیت شناسایی الگوریتم را با سرعت همگرائی سریعی کاهش خواهد داد. چگونگی افزایش توانایی شناسایی الگوریتم از طریق بهبود تنوع جمعیتی با اصلاح فرآیند بروز رسانی افراد برای بهبود ویژگی های کلی الگوریتم بهینه سازی بسیار مهم می باشد. از این رو، LBSA پیشنهادی عملکرد سراسری BSA بنیادی را توسط اصلاح فرآیند های جهشی بهبود می بخشد.

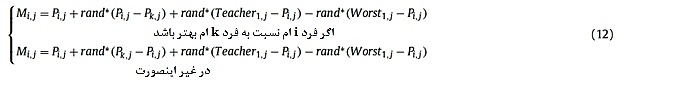
**4.2 جهش LBSA**

در BSA بنیادی، تنها از اطلاعات پیشین و موقعیت فعلی یک فرد برای تولید موقعیت جدید استفاده می شود. در این الگوریتم افراد نمی توانند دانش را از آموزش دهنده کل گروه یادبگیرند. برخی از روش های EA که می توانند بهترین فرد را ردیابی می کنند ممکن است سرعت همگرائی الگوریتم را افزایش دهند. برای بهبود توانایی یادگیری BSA، هدایت کننده یادگیری بهترین فرد را در فرآیند جهش الگوریتم معرفی می کند که به شرح زیر می باشد:



که آموزش دهنده بهترین فرد از نسل فعلی است، F=3\*rand و معانی پارامترهای دیگر همان چیزی هستند که در BSA بنیادی وجود دارد.

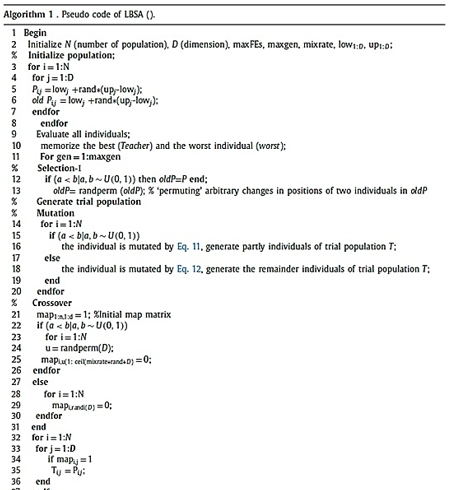
روش های یادگیری TBLO قبل از آنکه اصلاح شود در LBSA به کار رفته است . روش جدید شامل 3 بخش می باشد. بخش اول، هر فرد دانش را از یک فرد تصادفی و از جمعیت حاضر ، مانند همان چیزی که در روش TBLO است یاد می گیرد. بخش دوم، هر فرد دانش را از بهترین فرد، که متفاوت از TBLO است یاد می گیرد زیرا از موقعیت میانگین جمعیت حاضر استفاده نمی شود. علاوه بر این، برای افزایش تنوع افراد تولید شده، اجتناب از بدترین فرد نسل حاضر در معادله بروز رسانی درنظر گرفته می شود. در مقابل TBLO، سه روش ذکر شده در LBSA را در یک معادله یکپارچه کرده است و هر فرد تنها یک را در هر نسل کنترل می کند. جزئیات دقیق به شرح زیر می باشد:

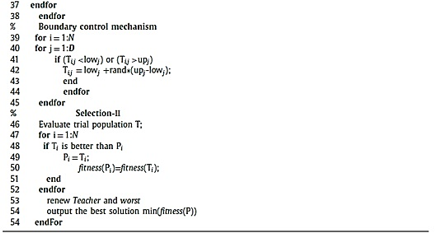


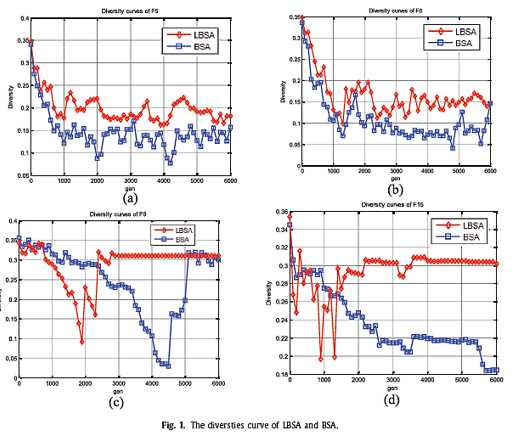
که بیت j ام از فرد i ام می باشد، i= 1 , 2 , . . . , N ، j=1 , 2 , . . . , D و مقدار بیت j ام از فرد k ام می باشد، که از جامعه بصورت تصادفی انتخاب شده و متفاوت از فرد i ام است. مقدار بیت j ام از فرد i ام است که بعداز عملگر جهش بکار می رود. علاوه براین، آموزش دهنده و بدترین فرد، به ترتیب بهترین و بدترین موقعیت از نسل حاضر می باشند. ما مشاهده کردیم که معادلات 11 و 12 در فرآیند جهش LBSA بصورت تصادفی انتخاب شده اند. شبه کد نوشته شده برای LBSA در الگوریتم 1 نشان داده شده است.

**4.3 تجزیه و تحلیل تنوع BSA و LBSA**

به منظور بررسی تنوع، BSA بنیادی با عملگرهای اصلاح شده بهبود یافته اند و توابع F5 , F6 , F9 و F15 در CEC2005 برای آزمایش تغییر تنوع به عنوان نسل های افزایشی استفاده شده اند. بسیاری از اندازه گیرهای گوناگون، مانند SPD، HPD [38] ، روش های مبتنی بر میانگین فاصله همینگ (Hamming) همه ی افراد، فاصله بین موقعیت همه ی افراد و موقعیت مرکزی جمعیت و آنتروپی [4] وجود دارند. به سادگی و بطور مستقیم تنوع BSA و LBSA تجزیه و تحلیل می شود، روشی ساده شده براساس متوسط فاصله همینگ همه افراد در این مقاله انتخاب شده است. برای مقایسه منصفانه تنوع الگوریتم های BSA و LBSA، فاصله بین دو فرد در محدوده [0 , 1] نرمال سازی می شود. تعریف تنوع جمعیت به شرح زیر می باشد:







تعریف 1 : فاصله همینگ بین فرد i ام و () و فرد ام () در رابطه 13 بیان شده است:



تعریف 2 : فاصله نرمال شده بین فرد i ام () و فرد ام () در رابطه 14 بیان شده است:



تعریف 3 : متوسط فاصله همینگ بین جمعیت در هر تکرار در رابطه 15 نشان داه شده است:



پارامترهای آموزش عبارتند از: مقدار جمعیت 50 نفر است، شرط پایان الگوریتم حداکثر تعداد تکرار ها است، مقدار تکرار 6000 است، ابعاد توابع 30 است و دو الگوریتم آزمایش شده بیش از 30 بار بطور مستقل از هم اجرا شده اند. تغییرات تنوع میانگین در شکل 1 نشان داده شده است و بیان می کند که تنوع میانگین LBSA نسبت به BSA بنیادی برای بسیاری از تکرارها بهتر است. این کار به LBSA کمک می کند تا براحتی از نقطه بهینه محلی دور شود. شکل 1 (الف) و (ب) نشان می دهد که تنوع LSBA تقریبا نسبت به BSA بهتر است، و تنوع دو الگوریتم در تمام مسیر تکامل مرتعش می باشد. شکل 1 (ج) و (د) نشان می دهند که تنوع LBSA نسبت به BSA در مرحله شروع تکامل بدتر می باشد اما نتایج مخالفی در آنافاز تکاملی بدست آمده است. شکل 1 (ج) و (د) نشان می دهند که تنوع الگوریتم ها در پایان تکامل برای الگوریتم هایی که به نقطه بهینه محلی یا سراسری می رسند تقریبا بدون تغییر هستند.

**5 . شبیه سازی آزمایش ها**

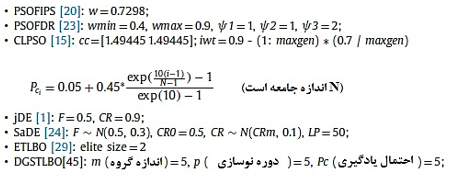
برای آزمایش عملکرد LBSA، سه آزمایش سنجشی انجام شده است. برای نشان دادن اثربخشی الگوریتم پیشنهادی، سه نوع از الگوریتم PSO (PSOFIPS [20] ، PSOFDR [23] و CLPSO [15])، دو نوع از الگوریتم DE (jDE [1] و SaDE [24]) سه نوع از الگوریتم TLBO که به تازگی پیشنهاد شده است (TLBO [31]، ETLBO [29] و DGSTLBO [45]) و BSA بنیادی [5] به عنوان الگوریتم های مقایسه ای استفاده شده است. الگوریتم های مرتبط با LBSA عبارتند از ایده های یادگیری در PSO و TLBO، اطلاعات پیشین در BSA بنیادی و عملگر ترکیب در DE می باشند. از این رو، این الگوریتم ها یا انواع آنها برای مقایسه با LBSA انتخاب شده اند. برای نشان دادن اثربخشی الگوریتم پیشنهادی، CEC2005 [35] و CEC2014 [16 , 18] به عنوان مسائل تابع سنجشی بکار رفته اند. آموزش شبکه های عصبی مصنوعی (ANN) برای پیش بینی سری های زمانی و مدلسازی سیستم غیرخطی به عنوان مسائل کاربردی استفاده شده است.

**5.1 راه اندازی آزمایشی**

اطلاعات دقیقی باتوجه به مسائل سنجشی برای CEC2005 در مقاله [35] ارائه شده است. 30 مسئله سنجشی برای CEC2014 در مقاله [16] ارائه شده است. سیستم های غیرخطی تک ورودی تک خروجی(SISO) و چند ورودی تک خروجی (MISO) [39] و دو مسئله ی پیش بینی سری های زمانی کلاسیک ( سری های زمانی آشوبی مکی-گلاس و سری های زمانی آشوبی باکس-جنکینز [33,34] ) در آزمایش ها پیش بینی شده اند. مدل پیش بینی پرسپترون چند لایه (MLP) و پارامترهای آن توسط EAهای مختلف آموزش داده شدند.

**5.2 تنظیمات پارامتر**

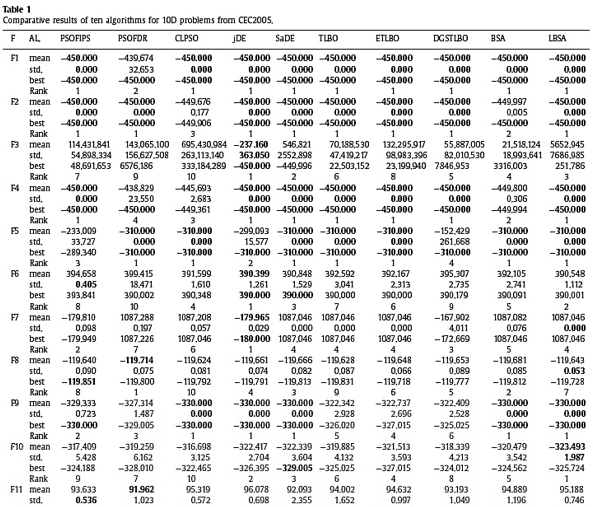
همه ی آزمایشات بر روی ماشینی مشابه با استفاده از برنامه MATLAB2012a انجام شده است. هر الگوریتم بطور مستقل برای 30 بار اجرا شده است. برای همه الگوریتم ها، مقدار جمعیت 50 نفر در نظر گرفته شده است. حداکثر به عنوان معیار توقف برای همه الگوریتم ها استفاده می شود. برای CEC2005 و CEC2014 ، حداکثر بر روی D 5000 (D ابعاد متغییر است) تنظیم شده است. علاوه بر این، توابع D 10 و D 30 برای CEC2005 شبیه سازی شده اند و توابع D 30 برای CEC2014 آزمایش شده اند. برخی از پارامترهای دیگر الگوریتم های مقایسه به شرح زیر ذکر شده اند:

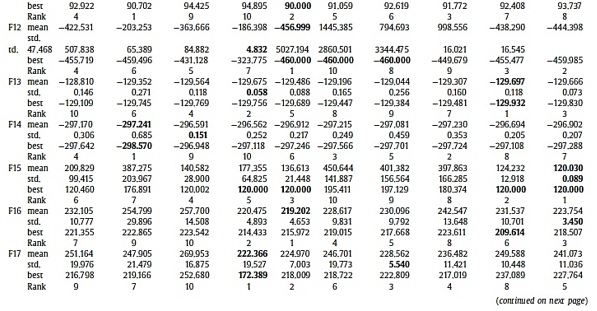


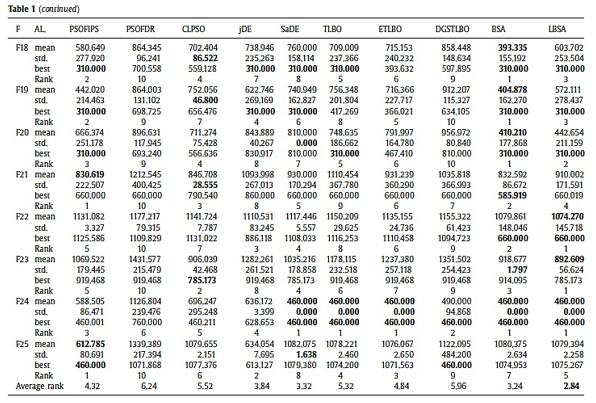
**5.3 آزمایش های CEC2005**

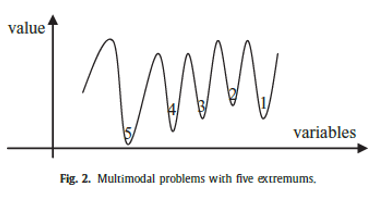
**5.3.1 مقایسه درستی راه حل تابع D 10 در CEC2005**

نتایج شبیه سازی برای تابع D10 در CEC2005 در جدول 1 نشان داده شده است. نتایج نشان می دهند که با توجه به مقدار میانگین، LBSA عملکرد بهتری نسبت به 9 الگوریتم دیگر برای توابع F9، F10، F15، F22 و F23 دارد. jDE عملکرد بهتری نسبت به نه الگوریتم دیگر برای توابع F3، F6، F7 و F17 دارد. برای تابع F3، مقدار میانگین jDE به راه حل بهینه سازی نظری نزدیکر است و انحراف معیار (Std) jDE در بین همه ی الگوریتم ها کوچکتر است. LBSA نمی تواند برای راه حل بهینه سازی نظری در همه ی 30 بار اجرا شدن همگرا شود، و مقدار میانگین و انحراف معیار آن نسبت به jDE و SaDE بدتر است. رتبه LBSA در شرایط میانگین 3 می باشد. SaDE نسبت به نه الگوریتم دیگر در شرایط میانگین برای دو تابع F12 و F16 عملکرد بهتری دارد. مقدار میانگین PSOFIPS نسبت به الگوریتم های دیگر برای توابع F21 و F25 بهتر می باشد. PSOFDR عملکرد بهتری نسبت به دیگر الگوریتم ها در شرایط میانگین برای توابع F8 ، F11 و F14 دارد. BSA نسبت به الگوریتم های دیگر برای توابع F13، F18 و F20 بهتر می باشد. برای تابع F20 ، بهترین راه حل های PSOFIPS، TLBO ، BSA و LBSA شبیه به راه حل بهینه سازی نظری می باشند؛ که بهینه سازی سراسری در 30 بار اجرا شدن این سه الگوریتم وجود دارد. برای تابع F1، همه الگوریتم ها به مقدار بهینه سراسری بجز برای PSOFDR همگرا می شوند. همه ی 10 الگوریتم تقریبا مقدار بهینه سراسری تابع F2 را بدست می آورند. برای تابع F4، PSOFIPS و CLPSO نمی توانند به مقدار بهینه سراسری همگرا شوند. برای تابع F5 ، سه الگوریتم نمی توانند مقدار بهینه سراسری را پیدا کنند. جدول 1 همچنین نشان می دهد که LBSA برای 10 تابع از 25 تابع در شرایط میانگین دارای رتبه اول است که نسبت به 9 الگوریتم دیگر بالاتر می باشد. رتبه متوسط LBSA نیز بهتر از ده الگوریتم برای CEC2005 می باشد. علاوه براین، در آن یک پدیده که راه حل بهترین میانگین و انحراف معیار یک الگوریتم با ویژگی مخالف است وجود دارد. برای مثال، انحراف معیار LBSA کمترین بود، اما مقدار میانگین برای تابع F16 بهترین نبود، و بدین معنی است که راه حل LSBA از 8 الگوریتم دیگر به نسبت بهتر بود اما انحراف معیار برای تابع F20 بهترین نبود. دلیل این پدیده به شرح زیر توضیح داده شده است. برای یک مشکل چندگانه (چند کیفیتی) با 5 نقطه اکسترمم درشکل 2، الگوریتم A و B برای یافتن بهینه سازی سراسری با اجرای m حالت مستقل استفاده می شود. اگر الگوریتم A برای نقطه 2 در m اجرا شود، الگوریتم B در نقطه 4 با m-1 اجرا می شود و در نقطه 5 باید یکبار اجرا شود. بهترین میانگین الگوریتم B از الگوریتم A بهتر خواهد بود، اما انحراف معیار الگوریتم A ممکن است نسبت به الگوریتم B کمتر باشد. در شرایط بهترین میانگین در حالت کلی عملکرد راه حل الگوریتم B از الگوریتم A بهتر است.





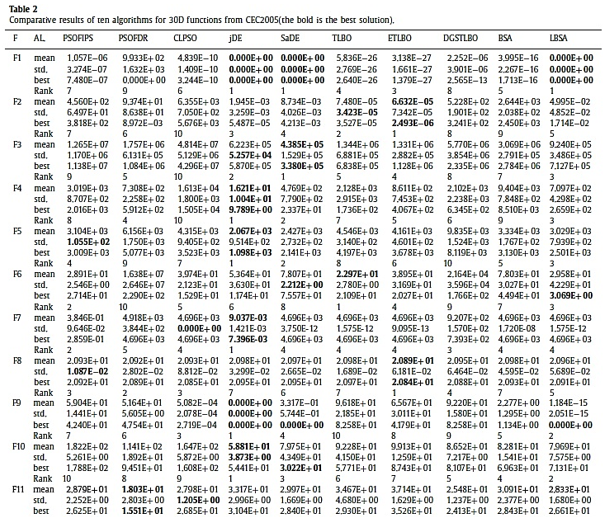


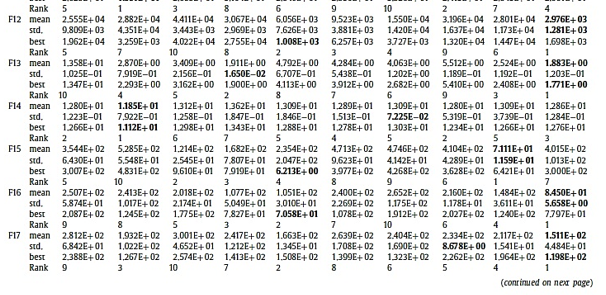


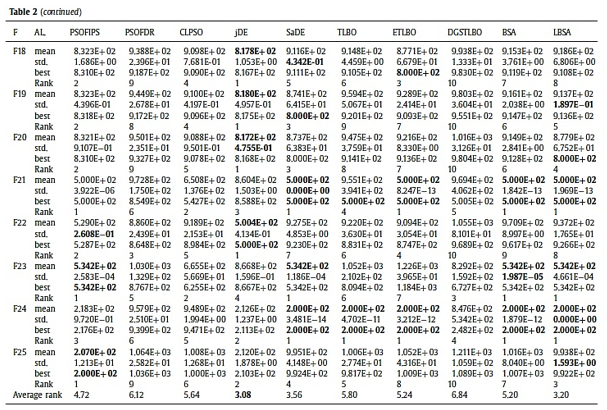
مقادیر برجسته در جدول 3 بهترین راه حل ها هستند. نتایج برای CEC2005 نشان می دهد که LBSA عملکردی قیاس پذیر را نتیجه می دهد، اگرچه همواره نسبت به دیگر روش ها در تمام جنبه ها بهتر نیست.

**5.3.2 مقایسه درستی راه حل تابع D 30 در CEC2005**

در این آزمایشات، تابع برازش برای تفاوت بین راه حل حقیقی الگوریتم ها و بهینه سازی نظری توابع انتخاب شده بود. اگر راه حل واقعی i ام الگوریتم باشد، پس تابع برازش می تواند بصورت بیان شود. ابعاد توابع درCEC2005 در این آزمایش 30 بود. بهترین، میانگین و انحراف معیار راه حل ها که بیش از 30 بار اجرا می شود برای 10 الگوریتم در جدول 2 نشان داده شده است. علاوه بر این، رتبه و رتبه میانگین 10 الگوریتم برای 25 تابع نیز داده شده است. جدول 2 نشان می دهد که رتبه LBSA در شرایط میانگین بهترین راه حل برای توابع F1، F12، F13، F16، F17، F21 ، F23، و F24 اول بود. jDE برای توابع F1، F4، F6، F9،F10، F18،F19، F20 و F22 رتبه اول شده است. اولین رتبه از دیگر توابع توسط الگوریتم های دیگر به اشتراک گذاشته شد. در شرایط کل تعداد رتبه های اول، jDE عملکرد بهتری نسبت به 9 الگوریتم دیگر و LBSA در مکان دوم بود. بهترین نتایج در جدول 2 بصورت برجسته نشان داده شده است. انحراف معیار LBSA نسبت به الگوریتم های دیگر برای توابع F12، F16،F19 ، F24 و F25 کوچکتر می باشد و الگوریتم jDE عملکرد بهتری نسبت به دیگر الگوریتم ها برای توابع F3، F4، F9، F10 ، F13 و F20 دارد. کوچکترین راه حل ها از نظر انحراف معیار برای دیگر توابع توسط هشت الگوریتم بدست می آید و جمع آنها برای هر الگوریتم کمتراز 6 بود. نتایج آماری در جدول 2 نشان می دهد که عملکردهای سراسری jDE و LBSA نسبت به دیگر الگوریتم ها بهتر هستند اگرچه jDE نسبت به LBSA در این آزمایش کمی بهتر بود. برای نمایش فرآیند همگرائی از 10 الگوریتم برای برخی از توابع در CEC2005، شکل 3 شامل هشت نمودار همگرائی برای D 30 توابع F1، F2 (توابع تک مدی)، F12، F13 (توابع چند مدی)، F16، F17، F23 و F24 (تابع ترکیبی) است. این شکل نشان می دهد که سرعت همگرائی SaDE نسبت به دیگر الگوریتم ها برای F1 سریع تر می باشد و ETLBO عملکرد بهتری نسبت به دیگر الگوریتم ها برای تابع F2 دارد. برای F12، F13، F16 و F17، LBSA عملکرد سریع تری دارد. برای F23، سرعت همگرائی SaDE، BSA و LBSA تقریبا مشابه هم هستند. برای F24، SaDE، TLBO، ETLBO، BSA و LBSA تقریبا سرعت همگرائی مشابه ای دارند. جدول 2 و شکل 3 همچنین نشان می دهند که LBSA عملکردی قابل مقایسه با الگوریتم های دیگر می دهد ، اما همیشه بهترین الگوریتم برای همه اندازه گیرهای آماری نیست. مقایسه عملکرد بین LBSA و jDE نشان می دهد که، ما می توانیم تشخیص دهیم که میانگین بهترین راه حل jDE برای 10 تابع کوچکترین بود، اما LBSA کوچکترین میانگین بهترین راه حل را برای هشت تابع ارائه می دهد. رتبه متوسط در شرایط میانگین بهترین راه حل jDE برابر با 3.08 بود که نسبت به مقدار (3.20) LBSA بهتر است. مقایسه نشان می دهد که jDE عملکرد بهتری نسبت به LBSA برای این 25 تابع دارد.

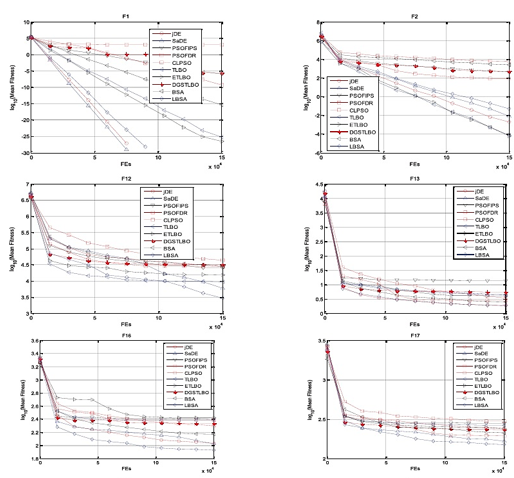


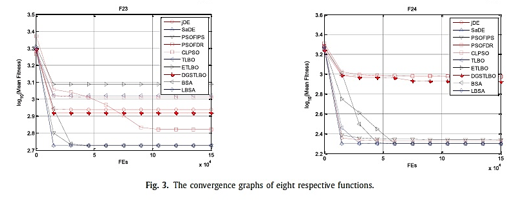




**5.3.3 مقایسه آزمون T برای D30 در CEC2005**

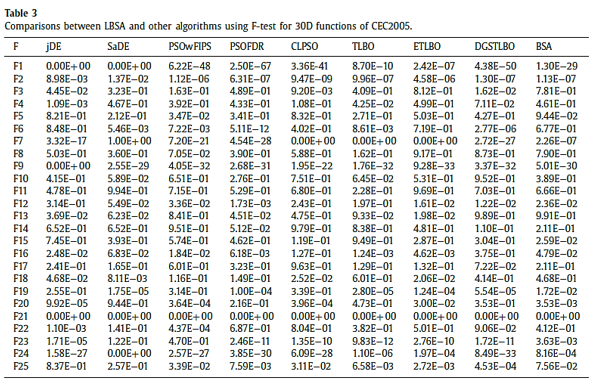
تفاوت های بین LBSA و دیگر الگوریتم ها با استفاده از آزمون t مورد آزمایش قرار گرفتند[42]. قبل از آزمون t، یک آزمون t برای آزمایش برابری واریانس بین LBSA و دیگر الگوریتم ها استفاده شده است. مقادیر F بین LBSA و دیگر الگوریتم ها در جدول 3 نشان داده شده است. جدول 3 نشان می دهد که تفاوت های واریانس بین LBSA و دیگر الگوریتم ها معنی دار نبودند. در آزمایش های آزمون t، آزمون دو دامنه ای (دو دنباله) با سطح معنی دار 0.05 در این مقابه انجام شده است و مقادیر t و مقادیر p برای هر جفت تابع در جدول 4 ذکر شده است. نتایج بهتر بین LSBA و الگوریتم های دیگر برای تفاوت توابع بصورت برجسته در شکل نشان داه شده اند. '' B '' ، '' W '' و '' S '' نشان می دهند که LBSA به ترتیب بطور معنی داری بهتر از، تقریبا مشابه، یا بطور معنی داری بدتر از الگوریتم مقایسه شده عمل می کنند. جدول 4 نشان می دهد که متوسط نرخ بسیار عالی و خوب بین LBSA و دیگر الگوریتم ها برای توابع D30 در CEC2005 در شرایط آزمون t برابر با 72.22 درصد () بود و متوسط عملکرد LBSA خوب بود، اگرچه همیشه بهترین الگوریتم برای همه توابع با توجه به 10 الگوریتم نبود. این جدول همچنین نشان می دهد که عملکرد BSA بنیادی توسط LBSA بهبود یافته است.





**5.4 آزمایشات برای توابع در CEC2014**

CEC2014 شامل 30 تابع سنجشی می باشد. از آنجا که بهینه سازی سراسری خود را منتقل می کنند، پیدا کردن آنها نسبتا دشوار است و راه حل های قابل قبول سخت برای بدست آوردن جواب دارند. در این بخش، تنها میانگین، انحراف معیار، بهترین راه حل ها و نتایج آزمون t حاصل از الگوریتم های مختلف داده می شود. در این بخش، تابع برازش خطای بین راه حل بهینه حقیقی و راه حل الگوریتم بهینه سازی است. فرض براین است که راه حل بهینه سراسری حقیقی و بهترین راه حل به دست آمده توسط الگوریتم های بهینه سازی باشد. سپس، برای تابع برازش انتخاب می شوند.



**5.4.1 مقایسه درستی راه حل تابع D10 در CEC2014**

نتایج آماری در ظرایط میانگین، انحراف معیار، بهترین راه حل، رتبه باتوجه به میانگین و متوسط رتبه الگوریتم های مختلف برای توابعD30 در CEC2014 در جدول 5 ذکر شده است. بهترین نتایج به صورت برجسته نشان داده شده اند. جدول 5 نشان می دهد که LBSA بهترین عملکرد را در شرایط میانگین برای توابع F26، F30، F34، F36، F37، F39،F41، و F47 دارد. jDE بهترین عملکرد را در شرایط میانگین برای توابع F28، F29، F33، F35، F45، F48، F50، F53 و F55 دارد. عملکرد در شرایط میانگین با SaDE نسبت به دیگر الگوریتم ها برای توابع F27، F31، F32، F42، F43،F44، F46 و F52 بهتر بود.برای تابع F38 ، BSA کوچکترین میانگین در بین تمام الگوریتم ها را دارا بود. برای تابع F40 ، میانگین FDRPSO کوچکترین بود. سه TLBO همان میانگین برای تابع F49 است که کوچکترین در 10 الگوریتم بود. برای تابع F54، PSOFIPS بهترین عملکرد در شرایط میانگین را دارد. باتوجه به بهترین راه حل ها، LBSA عملکرد بهتری نسبت به دیگر الگوریتم ها برای توابع F26، F29، F30، F34 ، F37 ، F41 و F42 دارد. jDE بهترین عملکرد برای 11 تابع (F27، F28، F33، F35، F43، F45، F48، F50، F51، F53 و F55)دارد و عملکرد در شرایط بهترین راه حل SaDE نسبت به دیگر الگوریتم ها برای توابع F27، F28، F31، F32، F42، F46، و F51 بهتر بود. PSOFIPS نسبت به دیگر الگوریتم ها در شرایط بهترین راه حل برای توابع F48، F51، و F54 بهتر است. PSOFDR عملکرد بهتری نسبت به دیگر الگوریتم ها برای توابع F26، F41، F42، F44، و F51 دارد. این جدول همچنین رتبه را باتوجه به میانگین 10 الگوریتم برای 30 تابع نشان می دهد. LBSA برای 9 تابع ، jDE برای 10 تابع و SaDE برای 9 تابع در مکان اول رتبه بندی شده اند. متوسط رتبه jDE و LBSA مشابه هم بودند. رتبه الگوریتم های دیگر نسبتا پایین تر از این سه الگوریتم بود.

**5.4.2 مقایسه آزمون t برای توابع در CEC2014**

نتایج آزمون t برای CEC2014 بین LBSO و دیگر الگوریتم ها در جدول 6 ذکر شده است، که نشان می دهد متوسط نرخ بسیار عالی و خوب بین LBSA و دیگر الگوریتم ها برای توابع D30 از CEC2014 در شرایط آزمون t برابر با 82.96 درصد () بود. جدول همچنین نشان می دهد که متوسط عملکرد LBSA خوب بود و عملکرد BSA بنیادی بهبود یافته است.

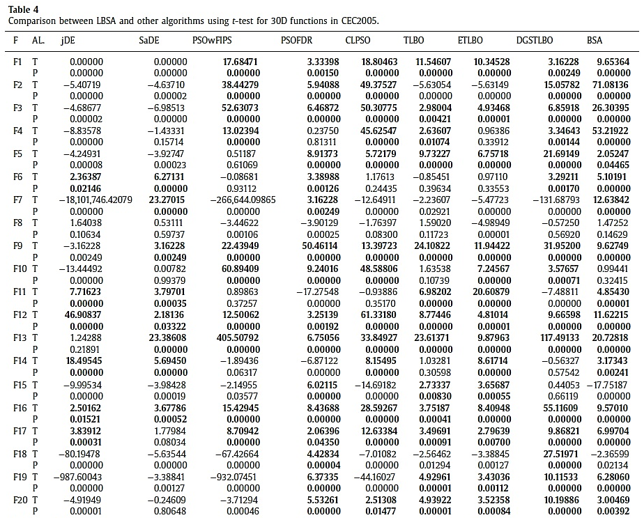
**6 . آموزش شبکه عصبی مصنوعی**

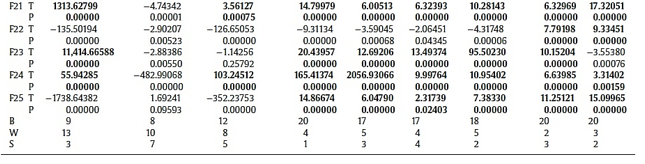
شناسایی سیستم روشی برای شناسایی یا اندازه گیری مدل ریاضی یک سیستم از اندازه گیری های ورودی و خروجی سیستم است. بدست آوردن مدل دقیق ریاضی از یک سیستم غیرخطی گاهی اوقات دشوار است و از این رو ایجاد یک مدل هوشمند بسیار مهم است. روش های پیش بینی L'' گام به جلو'' از حالات زودتر برای پیش بینی حالت استفاده می کند. می توان آن را به شرح زیر بیان کرد:

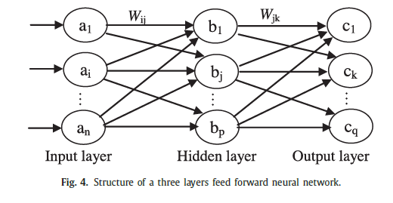


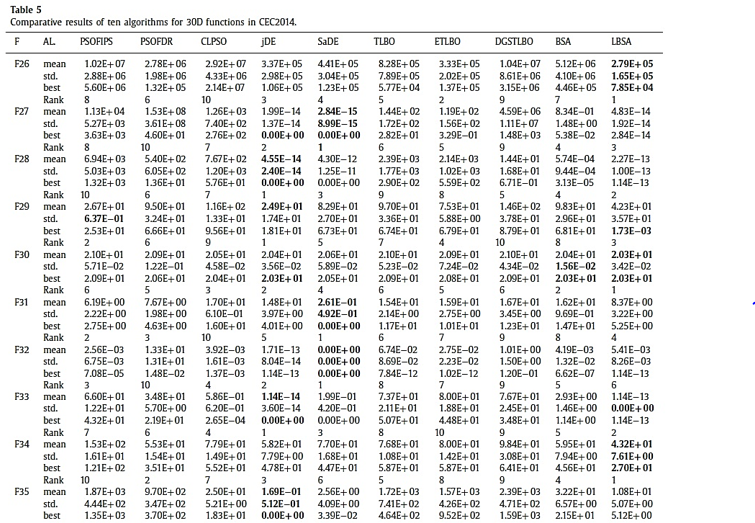
که در آن d تعداد متغییرهای ورودی یا بعد می باشد و L تعداد مراحل روبه جلو در مدل پیش بینی مستقیم'' چند گام به جلو'' است. یک ANN ساختار ریاضی انعطاف پذیری دارد که می تواند برای شناسایی سیستم های غیرخطی و پیش بینی سری های زمانی آشوبی استفاده شود. تکنیک بهینه سازی قدرتمند ممکن است به ANN کمک کند تا پارامترهای بهینه برای مسائل چند مدی و ابعاد بالا را پیدا کند. به نوبه خود، آموزش پارامترهای ANN معمولا به منظور آزمایش اثر الگوریتم بهینه سازی انجام می شود. برای آزمایش اثر LBSA، سه شبکه عصبی پیشسو (feed-forward) برای شناسایی سیستم غیرخطی و پیش بینی سری های زمانی آشوبی به عنوان اهداف یادگیری انتخاب شدند. سه لایه ANN () در شکل 4 نشان داده شده است.

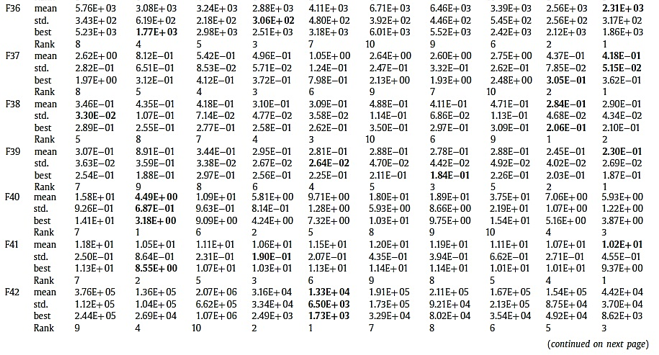
در شکل 4 ، a ، b و c واحدهای تغذیه یا بایاس هستند؛ وزن بین لایه ورودی و لایه پنهان است ؛ و وزن بین لایه پنهان و لایه خروجی می باشد ، .

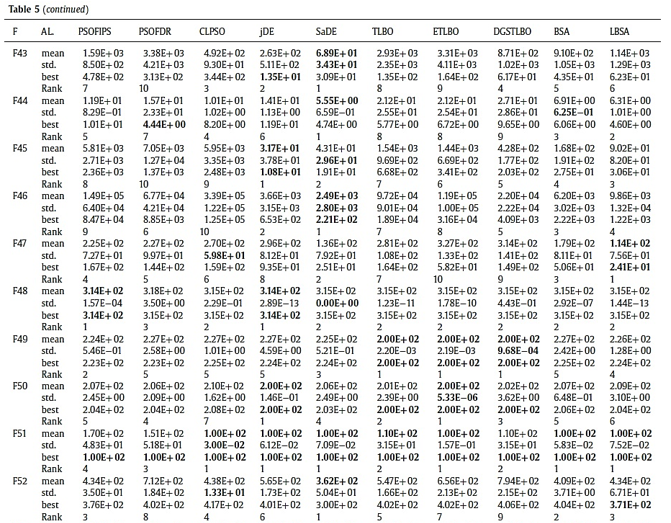


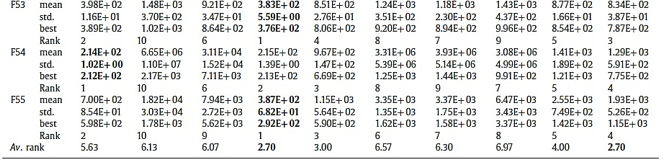








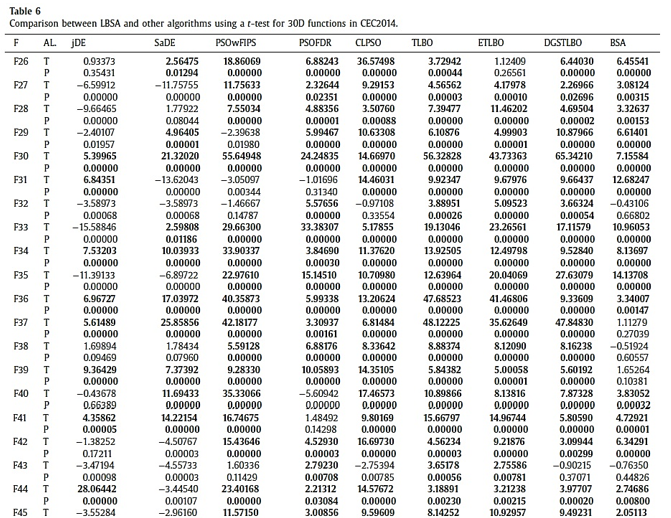


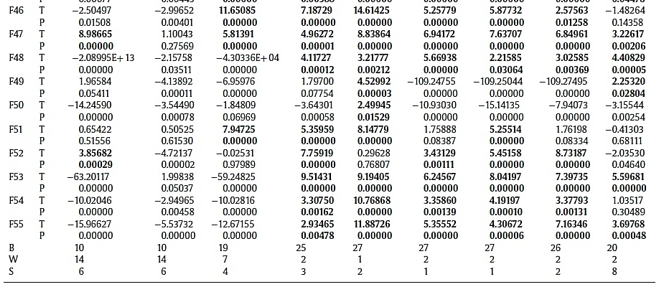


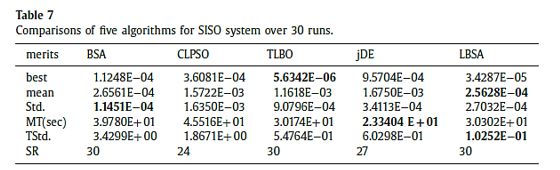
آموزش یک ANN به معنی بهینه سازی وزن و بایاس آن است. برای سه لایه ANN ، تعداد متغییرهایی که باید بهینه سازی شوند است و میانگین مجموع خطاهای مربع (MSE) در تمامی الگوهای آموزش اغلب به عنوان تابع هدف الگوریتم های بهینه سازی انتخاب می شود، محاسبه به شرح زیر می باشد:

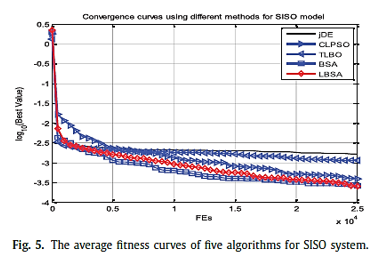


که m اندازه داده های آموزش، n تعداد خروجی ها، خروجی موردنظر و خروجی ANN می باشد.









**6.1 شناسایی سیستم غیرخطی**

**A . سیستم غیر خطی SISO**

سیستم غیرخطی SISO استفاده شده در این آزمایش به شرح زیر می باشد:



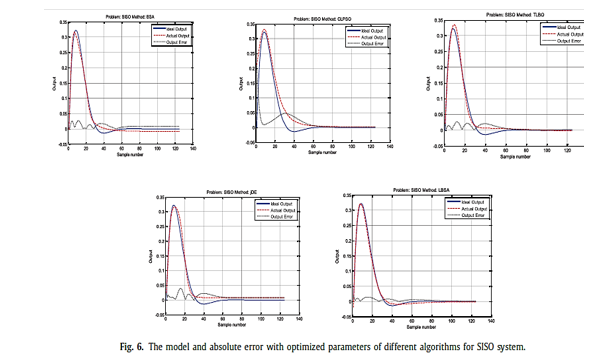
در این آزمایش، اندازه شبکه بود و تعداد متغییر هایی که باید تعیین شوند 16 بود. علاوه براین، 124 جفت از داده ها از معادله 18 به عنوان نمونه های آموزش انتخاب شده بود. نتایج به وضوح نشان می دهند ، تنها 5 الگوریتم بهینه سازی برای بهینه سازی شبکه عصبی استفاده شده بود. اندازه جمعیت همه ی الگوریتم ها 50 بود، حداکثر ، 25000 بود و دیگر پارامترهای الگوریتم های مختلف مشابه مورد CEC2005 استفاده شده است. جدول 7 نشان دهنده نتایج آموزش 5 الگوریتم برای اجرا شدن مستقل بیش از 30 بار می باشد. در اینجا، '' بهترین '' مقدار MSE است و '' میانگین'' متوسط بهترین مقدار از MSE می باشد، '' Std. '' انحراف معیار بهترین راه حل است و '' MT '' میانگین زمان همگرائی الگوریتم برای یک راه حل قابل قبول است. اگر الگوریتم به راه حلی بهینه در 30 بار اجرا شدن همگرا نشود، '' MT '' میانگین زمان الگوریتم است که توسط حداکثر به پایان می رسد. علاوه بر این، '' .TStd '' انحراف معیار زمان است و '' SR '' نسبت موفق 30 بار اجرا شدن مستقل برنامه می باشد. راه حل قابل قبول 0.002 است. جدول نشان می دهد که بهترین و میانگین MSE از LBSA کوچکترین بود و Std. در BSA کوچکترین مقدار از 5 الگوریتم بود. MT از jDE کوچکترین بود و Tstd. در LBSA کوچکترین بود. نسبت های موفق از 4 الگوریتم 100 درصد بود. فرآیند همگرائی باتوجه به در شکل 5 نشان داده شده است، که نشان می دهد سرعت همگرائی BSA نسبت به دیگر الگوریتم ها سریع تر بود و متوسط دقت همگرائی LBSA نسبت به دیگر الگوریتم ها بالاتر بود. نمونه های آموزشی الگوریتم های مختلف مشابه بودند. برای نشان دادن اثربخشی مدلسازی الگوریتم های مختلف، ANN ها با بهترین پارامترها شبیه سازی شدند. منحنی های مدلسازی با بهترین پارامترهای الگوریتم های مختلف در شکل نمایش داده شده و خطاهای مطلق نیز در منحنی های مربوطه برای الگوریتم نشان داده شده است. شکل 6 نشان می دهد که خطای آزمایش ANN با LBSA بهتر از چهار الگوریتم دیگر است.

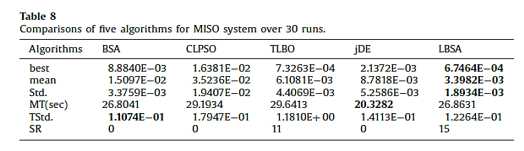
**B . سیستم غیرخطی MISO**

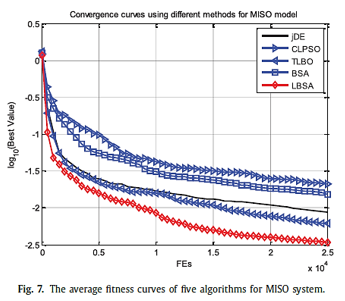
سیستم غیرخطی MISO استفاده شده در این آزمایش به شرح زیر می باشد:

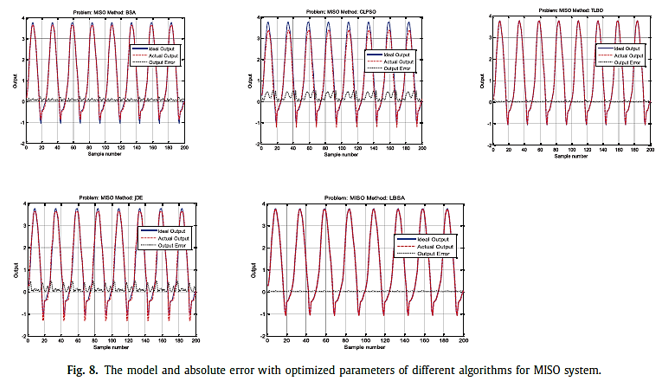


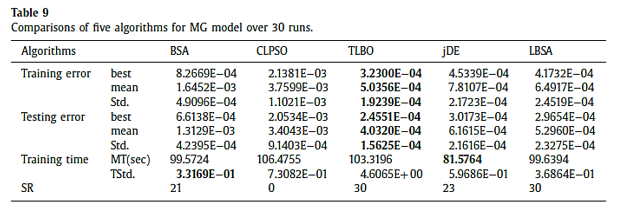
اندازه شبکه عصبی بود و تعداد متغییرهای الگوریتم های بهینه سازی 26 بود. علاوه براین، 200 جفت داده از رابطه 19 به عنوان نمونه های آموزشی انتخاب شده بودند. پارامترهای آموزش 5 الگوریتم مشابه سیستم SISO می باشند. MSE قابل قبول 0.002 می باشد. جدول 8 نتایج 5 الگوریتم مختلف را نشان می دهد. نتایج نشان می دهند که MSEها در شرایط بهترین، میانگین و Std. با LBSA کوچکترین در میان 5 الگوریتم بودند. زمان آموزش jDE کوتاه ترین بود TStd. در BSA کوچکترین بود. TLBO و LBSA میزان موفقیت بالاتری نسبت به دیگر الگوریتم ها داشتند. شکل 7 فرآیند همگرائی 5 الگوریتم را نشان می دهد. این شکل نشان می دهد که سرعت همگرائی LBSA نسبت به دیگر الگوریتم ها بیشتر است. منحنی های مدلسازی با بهترین راه حل ها الگوریتم های مختلفی را در شکل 8 نشان می دهند، که نشان می دهد خطاهای مطلق LBSA و TLBO از دیگر الگوریتم ها نسبتا کوچکتر است.











**6.2 پیش بینی سری های زمانی آشوبی**

**A . سری های زمانی آشوبی مکی-گلاس (MG)**

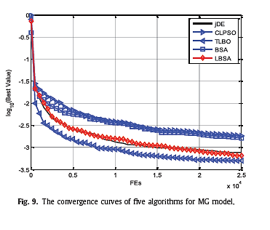
سیستم آشوبی مکی-گلاس [33,34] به شرح زیر است:

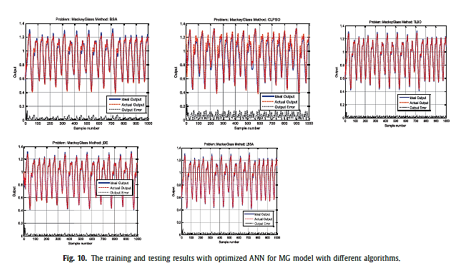


که ، و است. در حالت کلی، از چهار نقطه ی قبلی و برای پیش بینی نقطه بعدی استفاده می شوند. هدف از این آزمایش ایجاد مدل پیش بینی یک گام به جلوی MG است، همانطور که در رابطه 21 نشان داده شده است:



برای پیش بینی مدل MG ، ANN باید شامل 4 واحد ورودی و یک واحد خروجی و واحدهایی از لایه پنهان باشد که در این آزمایش 5 بود. مقدار جمعیت همه ی الگوریتم ها 50 بود، و حداکثر برابر 25000 بود. پارامترهای باقی مانده مشابه پارامترهای بکار رفته در بخش 6.1 بودند. 1000 جفت داده از مدل معادله 20 به عنوان آموزش و نمونه های آزمایش انتخاب شده بودند. در حالت قبل از 500 جفت داده به عنوان نمونه های آموزش استفاده شده بود و بقیه به عنوان نمونه های آزمایش انتخاب شده بودند. در حالت قبل 17 حالت بطور تصادفی تولید شدند. MSE به عنوان تابع برازش همه ی الگوریتم ها انتخاب شده بود و مقدار قابل قبول راه حل 0.002 است. اهمیت میانگین ، Std. ، نرخ موفقیت و زمان اجرای آموزش و فرآیند آزمایش در جدول 9 نشان داده شده اند. این نتایج از 30 بار اجرا شدن مستقل هر 5 الگوریتم بدست آمدند. جدول نشان می دهد که TLBO عملکرد بهتی نسبت به دیگر الگوریتم ها دارد. میانگین خطای آموزش و خطای آزمایش که با LBSA بدست آمده رتبه دوم از 5 الگوریتم را دارد. نرخ موفقیت TLBO و LBSA 100 درصد بود. زمان آموزش jDE کوچکترین بود و TStd. در BSA کوچکترین از 5 الگوریتم بود. این جدول همچنین نشلن می دهد که عملکرد LBSA نسبت به TLBO برای مدل MG بدتر بود. فرآیند همگرائی متوسط MSE برای همه ی الگوریتم ها در شکل 9 نشان داده شده است. این شکل نشان می دهد که سرعت همگرائی TLBO نسبت به دیگر الگوریتم ها برای مدل MG سریع تر است. LBSA رتبه دوم سریع ترین عملکرد را در آزمایش دارد. منحنی های آزمایش و آموزش برای 1000 نمونه در شکل 10 نشان داده شداند. نتایج نشان می دهد که دقت تقریبی TLBO و LBSA بالاتر می باشد.



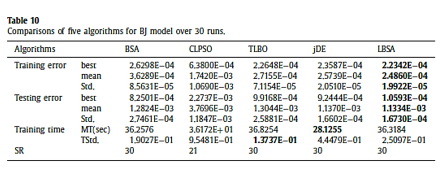


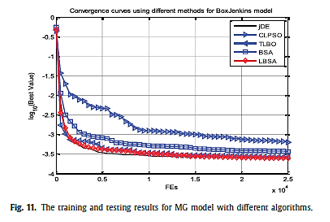
**B . سری های زمانی آشوبی باکس-جنکینز**

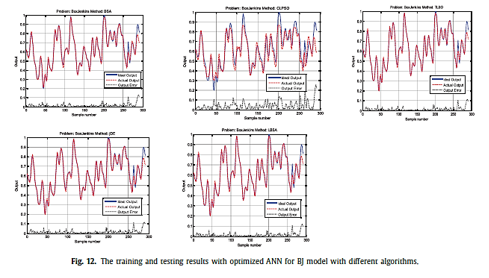
دومین سیستم آشوبی مدل کوره گاز باکس-جنکینز است. مجموعه داده های کوره گاز باکس – جنکینز از فرآیند احتراق مخلوط متان- هوا ثبت شده بود. در اینجا، 296 جفت داده از به عنوان نمونه های آموزش و آزمایش انتخاب شده بودند و همه ی مجموعه داده ها نرمالیزه شده بودند. علاوه براین، خروجی غلظت CO2 بود و ورودی دبی جریان گاز بود. علاوه بر این، و برای پیش بینی حالت بعدی مورد استفاده قرار گرفتند که به شرح زیر می باشند:



ساختار ANN دارای دو واحد ورودی، پنج واحد پنهان و یک واحد خروجی بود. 142 نقطه به عنوان نمونه های آموزش انتخاب شده بودند و 150 نقطه دیگر به عنوان نمونه های آزمایش مورد استفاده قرار گرفتتند. پارامترهای آموزش همه ی الگوریتم ها مشابه آزمایش مدل MG می باشد. نتایج همه ی الگوریتم ها در جدول 10 نشان داده شده است. شکل 11 نشان دهنده تغییرات در متوسط MSE برای 30 بار اجرا شدن مستقل 5 الگوریتم می باشد. نتایج آزمایش و آموزش با پارامترهای بهینه ANN برای الگوریتم های مختلف در شکل 12 نشان داده شده است. جدول 10 نشان می دهد که LBSA عملکرد بهتری نسبت به دیگر الگوریتم ها با توجه به خطاهای آموزش و آزمایش دارد. میانگین زمان آموزش jDE کوچکترین از بین تمام الگوریتم ها بود و TStd. در TLBO کوچکترین بود. چهار الگوریتم به یک راه حل قابل قبول همگرا شده اند. شکل 11 و 12 نشان می دهند که سرعت همگرائی LBSA نسبت به دیگر الگوریتم ها برای نمونه های آموزش و آزمایش عملکرد بهتری دارد.







**7. نتیجه گیری**

در این مقاله، BSA بنیادی به توسعه LBSA توسط یک روش هدایت کننده یادگیری که فرآیند جهش BSA را تغییر می دهد پرداخته است. یادگیری از بهترین فرد در معادله جهش برای بهبود سرعت همگرائی الگوریتم استفاده شده است، و تنوع الگوریتم توسط جلوگیری از بدترین فرد نسل حاضر بهبود یافته است. سه روش یادگیری در یک معادله یکپارچه شده اند. از مقایسه TLBO پی بردیم که تنها یک برای یک فرد در هر تکرار وجود دارد. چارچوب بنیادی BSA تغییر داده نشده است، و ویژگی ساده آن حفظ شده است. عملکرد LBSA در دو مجموعه آزمایش کلاسیک (CEC2005 و CEC214) آزمایش شده بود و مقایسه آن با نه الگوریتم دیگر نشان می دهد که LBSA عملکرد قابل مقایسه ای در بسیاری از موارد به دست می آورد. نتایج آزمایش برای مدلسازی سیستم غیرخطی و پیش بینی سری های زمانی آشوبی توسط ANN نیز نشان می دهد که LBSA عملکرد بیشتر مدل ها را به چالش می کشد. کارهای آتی باید روی استفاد کامل ساختار BSA توسط طراحی یک روش نگهداری تنوع قوی تر متمرکز شود زیرا LBSA نمی تواند بهینه سازی سراسری را برای برخی از توابع در CEC2005 و CEC2014 بدست آورد. علاوه بر این، گسترش دامنه کابرد LBSA جهت تحقیق مهم می باشد.

**سپاس گذاری**

بخشی از این مقاله توسط بنیاد ملی علوم طبیعی چین (شماره اعتبار 61572224 ، 61304082 و 41475017) و صندوق ملی علوم برای دانش پژوهان جوان برجسته (شماره اعتبار 61425009 ) حمایت شده است. بخشی از این مقاله توسط پروژه تحقیقات علوم طبیعی در استان آنهویی (شماره گرانت KJ2015ZD36)، بنیاد علوم طبیعی در دانشکده ها و دانشگاه استان آنهویی (شماره گرانت KJ2016A639) و پروژه همکاری های بین المللی علوم و فناوری استان آنهویی (شماره گرانت 10080703003) حمایت شده است.

نویسندگان از استاد سوگان سان برای ارائه کد PSO و ویرایش متن و نظراتش تشکر می کنند.

**References**

[1] J. Brest, S. Greiner, B. Boskovic, et al., Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput 10 (6) (2006) 646–657.

[2] Q. Cai, M.G. Gong, L.J. Ma, Greedy discrete particle swarm optimization for large-scale social network clustering, Inf. Sci. 316 (2015) 503–516.

[3] D.B. Chen, F. Zou, Z. Li, J.T. Wang, S.W. Li, An improved teaching-learning-based optimization algorithm for solving global optimization problem, Inf. Sci. 297 (2015) 171–190.

[4] D.B. Chen, C.X. Zhao, Particle swarm optimization with adaptive population size and its application, Appl. Soft. Comput. 9 (2009) 39–48.

[5] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.

[6] M. Ghasemi, M.M. Ghanbarian, S. Ghavidel, S. Rahmani, E.M. Moghaddam, Modified teaching learning algorithm and double differential evolution algorithm for optimal reactive power dispatch problem: a comparative study, Inf. Sci. 278 (2014) 231–249.

[7] M. Ghasemi, S. Ghavidel, M. Gitizadeh, E. Akbari, An improved teaching-learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow, Electr. Pow. Ener. Syst. 65 (2015) 375–384.

[8] M.G. Gong, Y. Wu, Q. Cai, Discrete particle swarm optimization for high-order graph matching, Inf. Sci. 328 (2016) 158–171.

[9] K. Guney, A. Durmus, S. Basbug, Backtracking search optimization algorithm for synthesis of concentric circular antenna arrays, Int. J. Antenn. Propag. 2014 (2014) Article ID: 250841, 11 pages, <http://dx.doi.org/10.1155/2014/250841>.

[10] B. Jürgen, H. Jan, Discrete particle swarm optimisation for ontology alignment, Inf. Sci. 192 (2012) 152–173.

[11] H.S. Keesari, R.V. Rao, Optimization of job shop scheduling problems using teaching–learning-based optimization algorithm, Opsearch 51 (4) (2013) 545–561.

[12] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Netw. 4 (1995) 1942–1948.

[13] X.L. Li, X.D. He, A hybrid particle swarm optimization method for structure learning of probabilistic relational models, Inf. Sci. 283 (2014) 258–266.

[14] J.Q. Li, Q.K. Pan, K. Mao, A discrete teaching–learning-based optimization algorithm for realistic flow shop rescheduling problems, Eng. Appl. Artif. Intell. 37 (2015) 279–292.

[15] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[16] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014, 2013, pp. 1–32. Technical Report.

[17] J. Lin, Oppositional backtracking search optimization algorithm for parameter identification of hyper chaotic systems, Nonlinear Dyn. (2014) 1–11.

[18] L.B. Ma, Y.L Zhu, Y. Liu, A novel bionic algorithm inspired by plant root foraging behaviors, Appl. Soft. Comput 37 (2015) 95–113.

[19] M.A. Medina, C.A.C. Coello, J.M. Ramirez, Reactive power handling by a multi-objective teaching learning optimizer based on decomposition, IEEE Trans. Power Syst. 28 (4) (2013) 3629–3637.

[20] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, Appl. Soft. Comput. 8 (3) (2004) 204–210.

[21] H.B. Ouyang, L.Q. Gao, X.Y. Kong, et al., Hybrid harmony search particle swarm optimization with global dimension selection, Inf. Sci (2016) 318–337 346–347.

[22] V. Patel, V. Savsani, Optimization of a plate-fin heat exchanger design throughan improved multi-objective teaching–learning based optimization (MO-ITLBO) algorithm, Chem. Eng. Res. Des. 92 (2014) 2371–2382.

[23] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: In Proc. Swarm Intelligence Symp., 2003, pp. 174–181.

[24] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[25] A. Ratnaweera, S. Halgamuge, H. Watson, Self-organizing hierarchical particle swarm optimizer with time varying accelerating coefficients, IEEE Trans. Evol. Comput 8 (3) (2004) 240–255.

[26] R.V. Rao, Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems, Decis. Sci. Letters 5 (2016), doi:10.5267/j.dsl.2015.9.003.

[27] R.V. Rao, V.D. Kalyankar, Multi-pass turning process parameter optimization using teaching–learning-based optimization algorithm, Scientia Iranica 20 (3) (2013) 967–974.

[28] R.V. Rao, V.D. Kalyankar, G. Waghmare, Parameters optimization of selected casting processes using teaching–learning-based optimization algorithm, Appl. Math. Model. 38 (2014) 5592–5608.

[29] R.V. Rao, V. Patel, An elitist teaching–learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (4) (2012) 535–560.

[30] R.V. Rao, V. Patel, An improved teaching–learning-based optimization algorithm for solving unconstrained optimization problems, Sci. Iran. Trans. D 20 (3) (2013) 710–720.

[31] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci 183 (1) (2012) 1–15.

[32] P.K. Roy, A. Sur, D.K. Pradhan, Optimal short-term hydro-thermal schedulingusing quasi-oppositional teaching learning based optimization, Eng. Appl. Artif. Intell. 26 (2013) 2516–2524.

[33] B. Samanta, Prediction of chaotic time series using computational intelligence, Expert Syst. Appl. 38 (9) (2011) 11406–11411.

[34] X. Song, X. Zhang, S. Zhao, L. Li, Backtracking search algorithm for effective and efficient surface wave analysis, J. Appl. Geophys. 114 (2015) 19–31.

[35] P.N. Suganthan, N. Hansen, J.J. Liang, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, 2005, pp. 1–50. Technical Report.

[36] C.Y. Tsai, C.J. Chen, A PSO-AB classifier for solving sequence classification problems, Appl. Soft. Comput. 27 (2015) 11–27.

[37] H.F. Wang, I. Moon, S.X. Yang, et.al, A memetic particle swarm optimization algorithm for multimodal optimization problems, Inf. Sci. 197 (2012) 38–52.

[38] H. Wang, H. Sun, C.H. Li, Diversity enhanced particle swarm optimization with neighborhood search, Inf. Sci. 223 (2013) 119–135.

[39] L. Wang, F. Zou, X.H. Hei, An improved teaching–learning-based optimization with neighborhood search for applications of ANN, Neurocomputing 143 (2014) 231–247.

[40] L. Wang, Y. Zhong, Y.Y. Zhao, W. Wang, et al., A hybrid backtracking search optimization algorithm with differential evolution, Math. Probl. Eng. 501 (2015) 769245.

[41] Z.L. Yang, K. Li, A. Foley, C. Zhang, A new self-learning TLBO algorithm for RBF neural modelling of batteries in electric vehicles, 2014 IEEE Cong. on Evol. Comput. (CEC), 2014 July 6-11.

[42] Z.H. Zhan, J. Zhang, Y. Li, et al., Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cy. B 39 (6) (2009) 1362–1381.

[43] C.J. Zhang, Q. Lin, L. Gao, Backtracking Search Algorithm with three constraint handling methods for constrained optimization problems, Expert Syst. Appl. 42 (2015) 7831–7845.

[44] L. Zhao, Y. Yang, PSO-based single multiplicative neuron model for time series prediction, Expert Syst. Appl. 36 (2) (2009) 2805–2812.

[45] F. Zou, L. Wang, X.H. Hei, et al., Teaching-learning-based optimization with dynamic group strategy for global optimization, Inf. Sci. 273 (2014) 112–131.