





دانشکده: مهندسی برق

گروه: الکترونیک

آموزش شبکه عصبی توابع پایه شعاعی به روش ترکیبی الگوریتم بهینه سازی توده ذرات و جستجوی

ممنوع برای تقریب توابع

دانشجو: محمد علی پور ورمزآبادی

استاد راهنما:

دکتر علی سلیمانی ایوری

پایان نامه جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۱

تقدیم به :

این اثر را به پدر و مادر عزیزم تقدیم می نمایم، که همواره در تمامی مراحل زندگی یار و یاور و پشتیبان من بوده اند.

## تشکر و قدردانی:

در ابتدا از استاد عزیزم، دکتر علی سلیمانی ایوری تشکر می نمایم، که در به انجام رساندن این امر همواره از دانش و راهنمایی های ایشان بهره مند بودم، همچنین از خانواده عزیزم تشکر می کنم، که در طول انجام این کار از هیچ گونه کمک و مساعدت دریغ نمودند. در انتها نیز از دوستان عزیزم که برخی قسمت های کار از کمک آن ها بهره مند شدم تشکر می نمایم.



## چکیده

تقریب توابع، از جمله مسائلی می باشد که از دیر باز مورد علاقه علوم مختلف، از ریاضیات کاربردی گرفته تا علوم کامپیوتر قرار دارد. همواره روش های متعددی برای تقریب توابع وجود داشته اند و با گذر زمان این روش ها مورد بهبود قرار گرفتند، و یا روش های جدیدی برای تقریب توابع معرفی شدند. یکی از روش های تقریب توابع، استفاده از شبکه های عصبی می باشد. آموزش شبکه های عصبی در واقع خود نوعی از تقریب توابع می باشد، چون در آن باید روابط حاکم بر لایه های مختلف شبکه را تقریب زد تا به خروجی مطلوب دست یابیم. آموزش شبکه عصبی عمدتاً به روش پس انتشار خطا یا گرادیان نزولی صورت می گیرد، اما گاهی از الگوریتم های ابتکاری<sup>۱</sup> نیز برای آموزش شبکه عصبی استفاده می شود که می تواند مزایا یا معایبی را به همراه داشته باشد. با توجه به کارهایی که تا کنون توسط پژوهشگران در این زمینه انجام شده، استفاده از الگوریتم های بهینه سازی به عنوان یک ابزار برای آموزش شبکه های عصبی می تواند زمان آموزش را کاهش دهد و میزان همگرایی را افزایش دهد و در نتیجه خطای کمتری را در پی داشته باشد.

از زمان معرفی الگوریتم های بهینه سازی همواره تلاش در ایجاد تغییرات و انجام ترکیب های مختلف در این الگوریتم ها به منظور بهبود پاسخ آن ها وجود داشته است، که باعث افزایش بازدهی، کاهش زمان جستجو و کاهش حجم محاسبات می شود، این تغییرات گاهی روی ساختار اصلی خود الگوریتم بهینه سازی صورت می پذیرد و گاهی با ترکیب این الگوریتم ها با سایر روش ها و الگوریتم های بهینه سازی انجام می گیرد.

---

<sup>۱</sup> Heuristic

در پایان نامه پیش رو، به منظور تقریب توابع از شبکه های عصبی توابع پایه شعاعی و برای آموزش آن از ترکیب الگوریتم بهینه سازی توده ذرات و الگوریتم جستجوی ممنوع استفاده کردیم، در واقع با استفاده از ترکیب این دو الگوریتم از مزایای دو الگوریتم استفاده کرده و معایب آن ها را کاهش می دهیم تا به پاسخ بهتر و دقت بالاتری دست یابیم. برای این منظور، انتخاب برخی پارامتر های الگوریتم بهینه سازی توده ذرات، را توسط الگوریتم جستجوی ممنوع انجام دادیم و تاثیر این کار را بر روی بازده آموزش شبکه عصبی توابع پایه شعاعی بررسی کردیم و با آموزش شبکه عصبی با روش گرادیان نزولی و آموزش با استفاده از الگوریتم بهینه سازی توده ذرات کلاسیک مورد مقایسه قرار دادیم.

کلمات کلیدی: شبکه عصبی، تقریب توابع، الگوریتم بهینه سازی توده ذرات، جستجوی ممنوع





## فهرست

و	چکیده
۲	۱. مقدمه
۸	۲. تقریب توابع و شبکه های عصبی
۹	۱.۲. تقریب توابع
۹	۱.۱.۲. معرفی:
۹	۲.۱.۲. تقریب چند جمله ای:
۱۰	۳.۱.۲. روش های تقریب چند جمله ای
۱۰	۱.۳.۱.۲. بسط تایلور
۱۱	۲.۳.۱.۲. پروجکشن
۱۱	۳.۳.۱.۲. درونیابی لاگرانژ
۱۲	۵.۳.۱.۲. درونیابی هرمیت
۱۲	۶.۳.۱.۲. چندجمله ای های متعامد
۱۴	۲.۲. شبکه های عصبی
۱۴	۱.۲.۲. تقریب توابع با استفاده از شبکه های عصبی مصنوعی:
۱۷	۲.۲.۲. شبکه های عصبی توابع پایه شعاعی:
۱۹	۳.۲.۲. الگوریتم گرادیان نزولی برای آموزش شبکه های عصبی توابع پایه شعاعی:
۲۰	۴.۲.۲. تقریب و درونیابی با استفاده از شبکه های عصبی توابع پایه شعاعی:
۲۱	۵.۲.۲. شبکه های عصبی موجک:
۲۴	۳. الگوریتم بهینه سازی توده ذرات
۲۵	۱.۳. معرفی

- ۲.۳. نسخه اصلی بهینه سازی توده ذرات: ..... ۳۰
- ۱.۲.۳. وزن اینرسی: ..... ۳۱
- ۲.۲.۳. ضریب تراکم: ..... ۳۲
- ۳.۲.۳. کاربردها: ..... ۳۲
- ۳.۳. دسته بندی بهینه سازی توده ذرات ..... ۳۴
- ۱.۳.۳. پیوستگی: ..... ۳۵
- ۲.۳.۳. مطابقت: ..... ۳۵
- ۳.۳.۳. وضعیت شناسی: ..... ۳۶
- ۴.۳.۳. محدودیت ها: ..... ۳۷
- ۵.۳.۳. ترکیب با روش های دیگر ابتکاری: ..... ۳۷
- ۴.۳. الگوریتم های بهبود یافته بر پایه بهینه سازی توده ذرات ..... ۳۸
- ۱.۴.۳. بهینه سازی توده ذرات با هدف فعال: ..... ۳۸
- ۲.۴.۳. بهینه سازی توده ذرات فعال: ..... ۳۸
- ۳.۴.۳. بهینه سازی توده ذرات با پیش سازی زاویه: ..... ۳۹
- ۴.۴.۳. الگوریتم بهینه سازی توده ذرات با بهترین چرخش: ..... ۳۹
- ۵.۴.۳. بهینه سازی توده ذرات باینری: ..... ۴۰
- ۶.۴.۳. بهینه سازی توده ذرات شبه تکاملی: ..... ۴۰
- ۷.۴.۳. بهینه سازی مقید با بهینه سازی توده ذرات: ..... ۴۰
- ۸.۴.۳. بهینه سازی توده ذرات پراکنده ساز: ..... ۴۱
- ۹.۴.۳. بهینه سازی توده ذرات با محدوده تقسیم شده: ..... ۴۱
- ۱۰.۴.۳. بهینه سازی توده ذرات قابل تنظیم و متغیر: ..... ۴۲
- ۱۱.۴.۳. بهینه سازی توده ذرات با همسایگی متغیر: ..... ۴۲

- ۴۳..... ۱۲.۴.۳. تخمین توزیع بهینه سازی توده ذرات:.....
- ۴۳..... ۱۳.۴.۳. بهینه سازی توده ذرات گسترده:.....
- ۴۳..... ۱۴.۴.۳. بهینه سازی توده ذرات گاوسی:.....
- ۴۴..... ۱۵.۴.۳. مدل بهینه سازی توده ذرات ژنتیک باینری:.....
- ۱۶.۴.۳. بهینه سازی توده ذرات حریرانه یا بهینه سازی توده ذرات با ساختار دوگانه کد شده
- ۴۴..... باینری:.....
- ۴۵..... ۱۷.۴.۳. بهینه سازی توده ذرات گروه دوستان:.....
- ۴۵..... ۱۸.۴.۳. بهینه سازی توده ذرات ابتکاری:.....
- ۴۶..... ۱۹.۴.۳. بهینه سازی توده ذرات با ساختار چند طبقه ای:.....
- ۴۶..... ۲۰.۴.۳. الگوریتم بهینه سازی توده ذرات مرکب با کاهش گرادیان:.....
- ۴۶..... ۲۱.۴.۳. بهینه سازی توده ذرات مرکب با تبرید شبیه سازی شده:.....
- ۴۷..... ۲۲.۴.۳. بهینه سازی توده ذرات مکرر:.....
- ۴۷..... ۲۳.۴.۳. بهینه سازی توده ذرات کاهش نقشه:.....
- ۴۷..... ۲۴.۴.۳. بهینه سازی توده ذرات باینری بهبود یافته:.....
- ۴۸..... ۲۵.۴.۳. بهینه سازی توده ذرات ژنتیک بهبود یافته:.....
- ۴۸..... ۲۶.۴.۳. بهینه سازی توده ذرات عصبی:.....
- ۴۸..... ۲۷.۴.۳. الگوریتم بهینه سازی جدید مبتنی بر بهینه سازی توده ذرات:.....
- ۴۹..... ۲۸.۴.۳. بهینه سازی توده ذرات جدید:.....
- ۴۹..... ۲۹.۴.۳. بهینه سازی توده ذرات نیچینگ:.....
- ۵۰..... ۳۰.۴.۳. بهینه سازی توده ذرات مرکب جدید:.....
- ۵۰..... ۳۱.۴.۳. بهینه سازی توده ذرات متعامد:.....
- ۵۰..... ۳۲.۴.۳. بهینه سازی توده ذرات موازی:.....

- ۳۳.۴.۳. بهینه سازی توده ذرات موازی غیرهمزمان: ..... ۵۱
- ۳۴.۴.۳. بهینه سازی توده ذرات موازی همزمان: ..... ۵۱
- ۳۵.۴.۳. بهینه سازی توده ذرات موازی مبتنی بر بردار: ..... ۵۱
- ۳۶.۴.۳. بهینه سازی توده ذرات با گسترش مناطق: ..... ۵۲
- ۳۷.۴.۳. بهینه سازی توده ذرات با رفتار فاصله ای: ..... ۵۲
- ۳۸.۴.۳. بهینه سازی توده ذرات با آشفستگی و صعود از تپه: ..... ۵۳
- ۳۹.۴.۳. بهینه سازی توده ذرات با سرعت فرار: ..... ۵۳
- ۴۰.۴.۳. بهینه سازی توده ذرات با گردآوری غیرفعال: ..... ۵۳
- ۴۱.۴.۳. بهینه سازی توده ذرات با گسترش فضایی ذرات: ..... ۵۴
- ۴۲.۴.۳. بهینه سازی توده ذرات با تعقیب و گریز: ..... ۵۴
- ۴۳.۴.۳. بهینه سازی توده ذرات مبتنی بر چاه پتانسیل کوانتوم دلتا: ..... ۵۵
- ۴۴.۴.۳. بهینه سازی توده ذرات تحت تاثیر کوانتوم: ..... ۵۵
- ۴۵.۴.۳. بهینه سازی توده ذرات با درون یابی چهارتایی: ..... ۵۶
- ۴۶.۴.۳. بهینه سازی توده ذرات با سرعت محصور: ..... ۵۶
- ۴۷.۴.۳. بهینه سازی توده ذرات با خود سازمان دهی: ..... ۵۶
- ۴۸.۴.۳. گونه در یک بهینه ساز توده ذرات: ..... ۵۷
- ۴۹.۴.۳. بهینه سازی توده ذرات آموزش دیده: ..... ۵۷
- ۵۰.۴.۳. بهینه سازی توده ذرات مبتنی بر دو توده: ..... ۵۷
- ۵۱.۴.۳. بهینه سازی توده ذرات غیر محدود: ..... ۵۸
- ۵۲.۴.۳. بهینه سازی توده ذرات متحد شده: ..... ۵۸
- ۵۳.۴.۳. بهینه سازی توده ذرات با همسایگی متغیر: ..... ۵۹
- ۵۴.۴.۳. بهینه سازی توده ذرات با سرعت محدود: ..... ۵۹

۵۹	.....	۵۵.۴.۳. بهینه سازی توده ذرات عمودی:
۶۱	.....	۴. جستجوی ممنوع
۶۲	.....	۱.۴. معرفی:
۶۳	.....	۲.۴. روش شناسی جستجوی ممنوع:
۶۳	.....	۳.۴. ویژگی های الگوریتم جستجوی ممنوع:
۶۴	.....	۱.۳.۴. لیست ممنوع:
۶۴	.....	۲.۳.۴. لیست نمایندگی:
۶۵	.....	۳.۳.۴. متمرکزسازی و غیر متمرکزسازی:
۶۵	.....	۴.۳.۴. تابع خطای هدف:
۶۶	.....	۵.۳.۴. شرایط توقف:
۶۶	.....	۴.۴. الگوریتم ممنوع:
۶۹	.....	۵.۴. نمونه هایی از ترکیب بهینه سازی توده ذرات و جستجوی ممنوع
		۱.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای مسئله ریزی کار و
۶۹	.....	تامین اجتماعی انعطاف پذیر
۷۰	.....	۲.۵.۴. بهینه سازی توده ذرات مبتنی بر ایده جستجوی ممنوع:
۷۱	.....	۳.۵.۴. الگوریتم بهینه سازی مرکب بهینه سازی توده ذرات و جستجوی ممنوع:
		۴.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای انتخاب ژن، برای کلاسه
۷۱	.....	بندی تومورها با استفاده از داده های تعریف ژن:
		۵.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای برنامه ریزی کارخانجات
۷۲	.....	فولاد مبتنی بر مدیریت معماری تولید - ذخیره و تولید - تقاضا:
		۶.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای بهینه سازی مسائل غیر
۷۳	.....	خطی:

۵. الگوریتم پیشنهادی ترکیب بهینه سازی توده ذرات و جستجوی ممنوع برای آموزش شبکه

۷۴

عصبی

۱.۵. تعریف مساله: ..... ۷۵

۲.۵. نسخه بهینه شده الگوریتم بهینه سازی توده ذرات: ..... ۷۷

۳.۵. ترکیب الگوریتم بهینه سازی توده ذرات بهینه شده و الگوریتم جستجوی ممنوع: ..... ۷۹

۴.۵. نتایج حاصل از الگوریتم پیشنهادی ..... ۸۶

۱۰۰

۶. بررسی و مقایسه نتایج

۱۱۰

۷. پیشنهادها و کارهای آینده

۱.۷. پیشنهادات و کارهای آینده ..... ۱۱۱

۲.۷. لیست مقالات استخراج شده: ..... ۱۱۳

چاپ شده: ..... ۱۱۳

در دست تدوین: ..... ۱۱۳

۱۱۴

۸. ضمایم

۱۲۹

Abstract

## فهرست جداول

جدول ۵-۱: بررسی زمان تقریبی آموزش با توجه به اندازه ابعاد توده ذرات..... ۹۲

جدول ۶-۱: خطا های نهایی حاصل از روش های مختلف برای آموزش شبکه عصبی برای توابع مورد

بررسی..... ۱۱۰

جدول ۶-۲: میزان همگرایی روش های بهینه سازی توده ذرات و الگوریتم پیشنهادی در طول

آموزش..... ۱۱۱

جدول ۶-۳: مقدار بهینه سازی شده ضرایب ۱ و ۲ برای آموزش توابع مختلف..... ۱۱۳

جدول ۸-۱: روابط توابع مورد بررسی ..... ۱۲۰

## فهرست شکل ها:

- شکل ۱-۲: ساختار شبکه عصبی توابع پایه شعاعی..... ۱۸
- شکل ۲-۲: موجک کلاه مکزیکی..... ۲۲
- شکل ۳-۲: موجک گاوسی..... ۲۳
- شکل ۴-۲: موجک مورلت..... ۲۳
- شکل ۱-۵: زیگزاگی شدن مسیر حرکت گرادیان در نزدیکی نقطه کمینه..... ۸۱
- شکل ۲-۵: فلوجارت الگوریتم پیشنهادی..... ۸۸
- شکل ۳-۵: فلوجارت زیربخش جستجوی ممنوع از الگوریتم پیشنهادی..... ۸۹
- شکل ۴-۵: رسم ۳ بعدی تابع RC..... ۹۳
- شکل ۵-۵: رسم ۳ بعدی تابع B..... ۹۳
- شکل ۶-۵: رسم ۳ بعدی تابع H..... ۹۴
- شکل ۷-۵: تقریب تابع AK با استفاده از الگوریتم پیشنهادی..... ۹۴
- شکل ۸-۵: تقریب تابع BI با استفاده از الگوریتم پیشنهادی..... ۹۵
- شکل ۹-۵: تقریب تابع BO با استفاده از الگوریتم پیشنهادی..... ۹۵
- شکل ۱۰-۵: تقریب تابع RC با استفاده از الگوریتم پیشنهادی..... ۹۶



- شکل ۵-۱۱: تقریب تابع CV با استفاده از الگوریتم پیشنهادی..... ۹۶
- شکل ۵-۱۲: تقریب تابع GP با استفاده از الگوریتم پیشنهادی..... ۹۷
- شکل ۵-۱۳: تقریب تابع G با استفاده از الگوریتم پیشنهادی..... ۹۷
- شکل ۵-۱۴: تقریب تابع HM با استفاده از الگوریتم پیشنهادی..... ۹۸
- شکل ۵-۱۵: تقریب تابع L با استفاده از الگوریتم پیشنهادی..... ۹۸
- شکل ۵-۱۶: تقریب تابع MT با استفاده از الگوریتم پیشنهادی..... ۹۹
- شکل ۵-۱۷: تقریب تابع RT با استفاده از الگوریتم پیشنهادی..... ۹۹
- شکل ۵-۱۸: تقریب تابع R با استفاده از الگوریتم پیشنهادی..... ۱۰۰
- شکل ۵-۱۹: تقریب تابع SC با استفاده از الگوریتم پیشنهادی..... ۱۰۰
- شکل ۵-۲۰: تقریب تابع SR با استفاده از الگوریتم پیشنهادی..... ۱۰۱
- شکل ۵-۲۱: تقریب تابع SS با استفاده از الگوریتم پیشنهادی..... ۱۰۱
- شکل ۵-۲۲: تقریب تابع T با استفاده از الگوریتم پیشنهادی..... ۱۰۲
- شکل ۵-۲۳: تقریب تابع Z با استفاده از الگوریتم پیشنهادی..... ۱۰۲
- شکل ۵-۲۴: تقریب تابع DJ با استفاده از الگوریتم پیشنهادی..... ۱۰۳
- شکل ۵-۲۵: تقریب تابع B با استفاده از الگوریتم پیشنهادی..... ۱۰۳

شکل ۵-۲۶: تقریب تابع DP با استفاده از الگوریتم پیشنهادی.....۱۰۴

١. مقدمه

## ۱. مقدمه

نیاز به تقریب توابع، ریشه در بسیاری شاخه های ریاضیات کاربردی و علوم کامپیوتر دارد. عموماً، مساله تقریب یک تابع از ما می خواهد، تابعی را در میان دسته ای از توابع تعریف شده انتخاب نماییم که تطبیق خوبی با تابع ورودی داشته باشد. تقریب توابع، که رابطه بین مجموعه ای محدود از داده های ورودی و خروجی را تعیین می کند، در محدوده بسیار وسیعی از کاربردهای جهان واقعی، مانند پیش بینی، بازشناسی الگو، استخراج داده و کلاسه بندی، مساله ای بنیادی و اساسی می باشد. روش های بسیاری برای حل این مساله ارائه شده اند که یکی از آن ها، شبکه های عصبی مصنوعی می باشد [۱].

شبکه های عصبی مصنوعی از جمله مباحثی است که در سال های اخیر دانشمندان علوم کامپیوتر به آن علاقمند شده اند و برای پیشرفت هرچه بیشتر علوم کامپیوتر وقت و هزینه بسیاری را صرف آن کرده و می کنند. این موضوع با ایده گرفتن از سیستم عصبی انسان و با هدف شبیه سازی هرچه بیشتر کامپیوتر به انسان شکل گرفت و در طول مدت زمانی که از معرفی آن می گذرد، پیشرفت بسیاری داشته است. شبکه های عصبی دارای انواع مختلفی می باشند و همچنین کاربرد های بسیاری دارند.

استفاده از یک شبکه عصبی در حل یک مساله، ساختاری شبکه ای را به کار می گیرد، به این مفهوم که، ارتباط میان لایه ها و نرون ها همانند معماری یک شبکه می تواند معماری ها و طراحی های مختلفی داشته باشد. عموماً تعیین ساختار یک شبکه عصبی خود مساله ای پیچیده است و به سادگی قابل حل نمی باشد. زمانی که تعداد لایه های قابل آموزش و واحد های پردازش بسیار کم باشند، شبکه توانایی آموزش مساله مورد نظر را ندارد. زمانی که تعداد لایه ها و نرون ها زیاد باشد، فرآیند آموزش بسیار طولانی خواهد بود. آموزش از روی نمونه ها بدین معنی است که شبکه با توجه به دانش حاصل از مجموعه نمونه ها توانایی یافتن تابع رابط بین ورودی و خروجی را دارا می باشد. این امر، بدین معنی است

که، پس از انجام گام آموزش، به ازای هر ورودی، با توجه به شرایط به دست آمده در گام آموزش مقدار خروجی متناظر با آن قابل پیش بینی خواهد بود. از این دیدگاه، آموزش مترادف با تقریب تابع می باشد.

یکی از انواع شبکه های عصبی، شبکه های عصبی توابع پایه شعاعی می باشند. توابع پایه شعاعی در ابتدا برای حل مسائل درونیایی چند متغیره حقیقی معرفی شده اند. برومهد<sup>۱</sup> و لو<sup>۲</sup> اولین اشخاصی بودند که از توابع پایه شعاعی در طراحی شبکه های عصبی استفاده کردند [۲].

لایه مخفی شبکه های عصبی توابع پایه شعاعی را می توان توسط الگوریتم های آموزشی ناظر<sup>۳</sup> آموزش داد. برای آموزش شبکه می توان یک الگوریتم مبتنی بر گرادینان نزولی را در نظر گرفت، هدف در این کار تعیین وزن های هر نرون در شبکه می باشد.

روش های استاندارد مختلفی برای آموزش شبکه های عصبی وجود دارد و همچنین روش های ابتکاری و فرا ابتکاری<sup>۴</sup> متعددی برای این منظور پیشنهاد داده شده اند که دارای نتایج مطلوبی می باشند. از این جمله می توان روش های بهینه سازی مانند الگوریتم بهینه سازی توده ذرات را نام برد.

الگوریتم بهینه سازی توده ذرات شباهت های بسیاری با روش های محاسباتی تکاملی مانند الگوریتم ژنتیک دارد. سیستم در ابتدا با مجموعه پاسخ های تصادفی مقاردهمی می شود و با به روزرسانی مرحله به مرحله به دنبال نقطه بهینه می گردد. این الگوریتم، یک الگوریتم مبتنی بر تکرار می باشد و با الهام از رفتار دسته پرندهگان و گروه ماهی ها و شبیه سازی رفتار آن ها معرفی گردید [۳].

---

<sup>۱</sup> Bromhead

<sup>۲</sup> Lowe

<sup>۳</sup> Supervised learning algorithms

<sup>۴</sup> Meta heuristic

ویژگی های این الگوریتم باعث شد تا به سرعت جای خود را در میان الگوریتم های بهینه سازی باز کند و در کاربردهای بسیاری به کار گرفته شود، و همین امر باعث اعمال تغییرات و بهینه سازی های مختلفی روی الگوریتم اصلی گردید تا بتوان از آن در کاربردهای گسترده تری بهره گرفت و همچنین بازدهی آن را افزایش داد و با توجه به کاربرد مورد نظر، به نتایج مطلوب تری دست یافت.

یکی از کاربرد های این الگوریتم استفاده از آن در آموزش شبکه های عصبی مصنوعی می باشد، که در این کاربرد، الگوریتم بهینه سازی توده ذرات، وظیفه تعیین وزن های لایه مخفی شبکه را بر عهده دارد و از واگرایی وزن ها جلوگیری می کند. همچنین کاربرد گسترده الگوریتم بهینه سازی توده ذرات در زمینه های گوناگون، باعث شده است تا در کاربردهایی که به طور مستقیم امکان استفاده از نسخه اصلی این الگوریتم وجود ندارد و یا به منظور دستیابی به دقت و بازده بیشتر و بالاتر، از این الگوریتم در ترکیب با سایر الگوریتم های بهینه سازی، مانند تبرید شبیه سازی شده، الگوریتم ژنتیک، جستجوی ممنوع، کلونی مورچگان و غیره استفاده کرد.

عموماً این ترکیب ها به منظور پوشش و کاهش ضعف الگوریتم ها در کاربرد مورد نظر صورت می گیرند، بنابراین با توجه به کاربرد مورد نظر، این ترکیب ها می تواند به نحوه های گوناگونی صورت گیرد. گاهی این الگوریتم ها به صورت موازی و در راستای یکدیگر روی مساله مورد نظر پیاده سازی می شوند و گاهی نیز دو الگوریتم به صورت پیچیده در هم ترکیب می شوند تا نواقصی را بر طرف سازند. این گونه ترکیب الگوریتم های بهینه سازی در صورتی که به خوبی صورت گیرد، می تواند با استفاده از نقاط قوت الگوریتم های مختلف و حذف و یا کاهش معایب و کاستی های این الگوریتم ها، باعث بهبود در پاسخ های نهایی شود. نحوه ترکیب های مختلف و همچنین ترکیب پیشنهادی و مزایای آن در ادامه مورد بررسی قرار خواهد گرفت.

در پایان نامه پیش رو، آموزش شبکه عصبی به منظور تقریب توابع، با استفاده از الگوریتم بهینه سازی توده ذرات که با الگوریتم جستجوی ممنوع ترکیب شده است صورت گرفته است.

الگوریتم جستجوی ممنوع نیز یکی از محبوب ترین الگوریتم های بهینه سازی می باشد، که مانند الگوریتم بهینه سازی توده ذرات یک الگوریتم مبتنی بر تکرار می باشد و از ساختار های حافظه ای بهره می گیرد. وجود این ساختار های حافظه ای به الگوریتم کمک می کند تا از تکرار پاسخ های بررسی شده و گیر کردن در کمینه های محلی جلوگیری کند و بازدهی الگوریتم را بهبود دهد. البته این الگوریتم دارای معایبی نیز می باشد که از آن جمله می توان کندی سرعت همگرایی الگوریتم جستجوی ممنوع در مقابل برخی دیگر از الگوریتم های بهینه سازی و عدم امکان استفاده از آن در مسائلی که دارای تعداد پارامترهای زیاد هستند را بر شمرد.

با توجه به معایب الگوریتم جستجوی ممنوع، می توان از این الگوریتم در ترکیب با الگوریتم بهینه سازی توده ذرات استفاده کرد، تا بتوانیم به الگوریتم مرکب بهتری نسبت به الگوریتم اصلی دست یابیم. هرچند که عموماً در ترکیب دو الگوریتم میزان محاسبات در هر تکرار افزایش می یابد، می توان با انتخاب نحوه درست ترکیب، تعداد تکرار های نهایی را کاهش داد و در نهایت میزان محاسبات را کاهش و سرعت همگرایی و بازده بهینه سازی را افزایش داد.

در پایان نامه پیش رو، به مطالعه الگوریتم های بهینه سازی پرداخته ایم و روش مناسبی را برای ترکیب دو الگوریتم بهینه سازی توده ذرات و جستجوی ممنوع برگزیده ایم تا بتوانیم از آن در آموزش شبکه عصبی توابع پایه شعاعی به منظور تقریب توابع بهره بگیریم و به نتایج مطلوب تری نسبت به سایر روش ها دست یابیم. سپس نتایج حاصل از پیاده سازی روش پیشنهادی برای تقریب توابع، با نتایج حاصل از

روش های متداول تقریب توابع مانند آموزش شبکه عصبی به روش گرادیان نزولی و آموزش شبکه عصبی با استفاده از الگوریتم بهینه سازی توده ذرات کلاسیک مورد بررسی و مقایسه قرار می دهیم.

حال به بررسی ساختار فصل بندی ادامه این پایان نامه می پردازیم.

در فصل ۱ به مروری کلی و معرفی جوانب مختلف این پایان نامه پرداختیم، تا به یک آشنایی کامل در مورد تعریف مساله و معرفی مواد اولیه مورد نیاز، که از آن ها در حل مساله بهره گرفته ایم دست یابیم.

در فصل ۲ در ابتدا تقریب توابع را تعریف کرده و لزوم انجام تقریب توابع را بیان می کنیم، سپس به بررسی روش های ریاضیاتی مختلف در تقریب توابع می پردازیم. در ادامه به معرفی شبکه های عصبی می پردازیم و انواع مختلف آن را بیان کرده و شبکه های عصبی توابع پایه شعاعی را بررسی می کنیم و نحوه استفاده از آن در تقریب توابع را بیان می کنیم.

در فصل ۳، به معرفی الگوریتم بهینه سازی توده ذرات می پردازیم، خاستگاه این الگوریتم را بیان کرده و خود الگوریتم را به تفسیر مورد شرح و بسط قرار می دهیم. در ادامه به بررسی بهینه سازی مختلف روی الگوریتم بهینه سازی توده ذرات می پردازیم.

در فصل ۴، به معرفی الگوریتم بهینه سازی جستجوی ممنوع می پردازیم، نحوه عملکرد و الگوریتم آن و اجزای تشکیل دهنده آن را معرفی کرده و وظایف هر یک را شرح می دهیم. سپس چند نمونه از ترکیب الگوریتم جستجوی ممنوع با الگوریتم بهینه سازی توده ذرات مورد بررسی قرار می دهیم.

در فصل ۵، الگوریتم پیشنهادی در ترکیب الگوریتم جستجوی ممنوع و الگوریتم بهینه سازی توده ذرات را معرفی می کنیم و نحوه استفاده از آن در آموزش شبکه عصبی توابع پایه شعاعی را به منظور تقریب توابع بیان می کنیم و نتایج حاصل از آن را بیان می کنیم.



در فصل ۶، نتایج به دست آمده از الگوریتم پیشنهادی را با نتایج حاصل از سایر الگوریتم ها مقایسه می کنیم و به بررسی و ارزیابی نتایج حاصل می پردازیم، و پیشنهادات و ایده های کارهای آتی را بیان می کنیم.

از آنجایی که الگوریتم بهینه سازی توده ذرات و الگوریتم جستجوی ممنوع به عنوان پایه های کار انجام گرفته می باشند، لذا به طور کامل تر مورد تفسیر و بررسی قرار گرفته اند و در نهایت چون الگوریتم پیشنهادی مبتنی بر الگوریتم های بهینه سازی توده ذرات و جستجوی ممنوع می باشد، از تکرار مطالب در فصل ۵، خودداری شده و صرفاً به نحوه ترکیب دو الگوریتم پرداخته ایم.

همچنین نحوه استفاده از الگوریتم بهینه سازی توده ذرات و الگوریتم گرادیان نزولی در آموزش شبکه عصبی در فصل ۶، آورده شده است، تا الگوریتم های آموزش را در کنار هم بهتر بتوان مورد بررسی قرار داد.

در فصل ۷، نیز پیشنهاداتی برای ادامه کار معرفی و مورد بررسی قرار می گیرد.

۲. تقریب توابع و شبکه های عصبی

## ۱.۲. تقریب توابع

### ۱.۱.۲. معرفی

نیاز به تقریب توابع، از بسیاری از شاخه های ریاضیات کاربردی و علوم کامپیوتر نشات می گیرد. عموماً، مساله تقریب یک تابع از ما می خواهد، تابعی را در میان دسته ای از توابع تعریف شده انتخاب نماییم که تطبیق خوبی با تابع ورودی داشته باشد.

می توانیم دو حالت عمده برای تقریب توابع در نظر بگیریم: در حالت اول، برای توابع هدف معین، نظریه تقریب توابع، شامل تحلیل و محاسبات عددی می شود، که تعیین کننده دقت و بازده تقریب، در تعیین نوع و دسته تابع می باشد. در حالت دوم، تابع هدف ناشناخته می باشد و به جای رابطه ریاضی، تنها مجموعه ای از داده و اطلاعات، به عنوان ورودی در اختیار داریم، به عنوان مثال می دانیم مقدار بعدی تابع، تابعی از مقدار فعلی است اما نمی دانیم این مقدار از چه رابطه ای تبعیت می کند.

در برخی موارد، ممکن است مقدار تابع، با خروجی بسیاری از توابع پیچیده یکسان باشد یا در واقع، توابع پیچیده بسیار زیادی وجود دارند که به ازای یک مقدار ورودی، خروجی یکسانی به ما می دهند و یافتن تنها یکی از این توابع، می تواند نیازمند زمان زیادی باشد، اما با یک تقریب، مقدار توابع بسیار سریعتر به دست می آیند.

### ۲.۱.۲. تقریب چند جمله ای

برای این منظور تابع را به صورت چند جمله ای با ضرایب نامعلوم مانند رابطه (۲-۱) در نظر می گیریم [۴]، اما در نهایت می بینیم که با انتخاب توابع مختلف، پاسخ های مختلفی خواهیم داشت، به همین

دلیل عموماً چند جمله ای را به صورت مجموعی از توابع مختلف با ضرایب مختلف به صورت رابطه (۲-۲) می نویسیم.

$$y_t = a. + a_1x + a_2x^2 + \dots + a_nx^n \quad (1-2)$$

$$y_t = a. + a_1T_1(x) + a_2T_2(x) + \dots + a_nT_n(x) \quad (2-2)$$

به عنوان مثال می توانیم چند جمله ای را به صورت مجموعی از توابع لگاریتمی در نظر بگیریم. در اکثر موارد فقط با یک تابع می توان تقریب را انجام داد، یا به عنوان مثال، اگر بدانیم تمامی مقادیر تابع مثبت هستند، می توانیم آنها را به صورت تابع نمایی در نظر بگیریم، مانند رابطه (۳-۲).

$$y_t = \exp(a. + a_1T_1(x) + a_2T_2(x) + \dots + a_nT_n(x)) \quad (3-2)$$

### ۳.۱.۲. روش های تقریب چند جمله ای

در اینجا چهار روش برای انجام تقریب چند جمله ای بیان می کنیم. رویه ها با توجه به استفاده از اطلاعات محلی مختلف و این که شامل اطلاعات مشتق می باشند یا نه، با هم تفاوت دارند.

#### ۱.۳.۱.۲. بسط تیلور

اگر اطلاعات شامل مقادیر تابع و مقادیر مشتق های مراتب بالای آن در یک نقطه باشد، آنگاه می توان تقریب تابع را با استفاده از بسط تیلور با رابطه (۴-۲) انجام داد [۵].

$$f(x) \approx f(x.) + (x - x.) \frac{\partial f(x)}{\partial x} \Big|_{x=x.} + \dots + \frac{(x - x.)^n}{n!} \frac{\partial^n f(x)}{\partial x^n} \Big|_{x=x.} \quad (4-2)$$

## ۲.۳.۱.۲. پروجکشن<sup>۱</sup>

متداول ترین حالت ممکن برای تقریب یک تابع اینگونه است که اطلاعات شامل مقادیر تابع در نقاط مختلف باشد. در این حالت دو رویه برای به دست آوردن ضرایب تابع چندجمله ای وجود دارد. روش اول انجام یک درونیابی می باشد [۵].

## ۳.۳.۱.۲. درونیابی لاگرانژ

این روش تقریب با رابطه (۲-۵) بیان می شود. ضرایب لاگرانژ، چند جمله ای هایی هستند که از رابطه (۲-۶) به دست می آیند. از آنجا که ضرایب لاگرانژ خود چند جمله ای می باشند، بنابراین تابع تقریب نیز یک چند جمله ای می باشد. با توجه به رابطه موجود برای ضرایب لاگرانژ، برای یافتن ضریب در یک نقطه رابطه (۲-۷) برقرار است.

$$f(x) \approx f.L.(x) + \dots + f_n L_n(x) \quad (۲-۵)$$

$$L_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (۲-۶)$$

$$L_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x \in \{x_0, \dots, x_n\} \setminus \{x_i\} \end{cases} \quad (۲-۷)$$

این روش برای نوشتن تابع چندان مفید نمی باشد اما نوشتن تابع به این صورت برای جمع عددی مفید می باشد [۵].

---

<sup>۱</sup> Projection

### ۵.۳.۱.۲. درونیابی هرمیت

دو این روش فرض می کنیم علاوه بر مقادیر تابع، مقادیر مشتق اول آن را نیز در همان نقاط داریم، در این صورت می توانیم تابع را با کمک رابطه (۸-۲) تقریب بزنیم [۵].

$$f(x) \approx \sum_{i=0}^n f_i H_i(x) + \sum_{i=0}^n \dot{f}_i \tilde{H}_i(x) \quad (۸-۲)$$

$$H_i = (1 - 2L'_i(x_i)(x - x_i))L_i(x)^2 \quad (۹-۲)$$

$$\tilde{H}_i = (x - x_i)L_i(x)^2 \quad (۱۰-۲)$$

$$H_i(x) = \tilde{H}'_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x \in \{x_0, \dots, x_n\} \setminus \{x_i\} \end{cases} \quad (۱۱-۲)$$

$$\tilde{H}_i(x) = H'_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x \in \{x_0, \dots, x_n\} \setminus \{x_i\} \end{cases} \quad (۱۲-۲)$$

### ۶.۳.۱.۲. چندجمله ای های متعامد

مشخص است که یافتن ضرایب چند جمله ای، همانند یک درونیابی، مشابه یک تابش روی فضای مورد نظر با استفاده از تابع می باشد. چند جمله ای های متعامد مانند توابع پایه ای هستند که ساختاری متعامد با اندازه ای متناسب دارند. توابع متعامد از رابطه (۱۳-۲) تبعیت می کنند.

$$\int_a^b T_i(x) T_j(x) \omega(x) dx = 0 \quad \forall i, j \ni i \neq j \quad (۱۳-۲)$$

توابع متداول متعامد، توابع چبیشف هستند، دلیل این محبوبیت نیز به وضوح مشخص است. چند جمله ای های چبیشف روی کران ۱- تا ۱ کراندار هستند و تابع وزن دهی آن ها نیز طبق رابطه (۱۴-۲) تعیین می شود.

$$\omega(x) = \frac{1}{(1-x^2)^{1/2}} \quad (14-2)$$

توابع پایه چیشف نیز با روابط (۱۵-۲، ۱۶-۲، ۱۷-۲، ۱۸-۲) تعیین می شوند.

$$T^c(x) = 1 \quad (15-2)$$

$$T_1^c(x) = x \quad (16-2)$$

$$T_{i+1}^c(x) = 2xT_i^c(x) - T_{i-1}^c(x) \quad i > 1 \quad (17-2)$$

$$f(x) \approx \sum_{j=0}^n a_j T_j^c(x) \quad (18-2)$$

فرم استاندارد چند جمله ای ها به صورت رابطه (۱۹-۲) می باشد. این امر در مورد چند جمله ای های متعامد نیز صدق می کند، بنابراین دلیل استفاده از چند جمله ای های متعامد این است که، در این صورت یافتن ضرایب چندجمله ای ساده تر خواهد بود، به عنوان مثال، اگر مرتبه بالاتری داشته باشیم، شرایط اولیه خوبی برای تقریب ضرایب مرتبه های پایین تر داریم.

$$b_0 + b_1x + \dots + b_nx^n \quad (19-2)$$

چندجمله ای های چیشف روی کران خاص تعریف می شوند [۵].

## ۲.۲. شبکه های عصبی

### ۱.۲.۲. تقریب توابع با استفاده از شبکه های عصبی مصنوعی

تقریب توابع، که رابطه بین مجموعه ای محدود از داده های ورودی و خروجی را تعیین می کند، در محدوده بسیار وسیعی از کاربردهای جهان واقعی، مانند پیش بینی، بازشناسی الگو، استخراج داده و کلاسه بندی، مساله ای بنیادی می باشد. روش های بسیاری برای حل این مساله ارائه شده اند که یکی از آن ها، شبکه های عصبی مصنوعی می باشد.

آموزش یک نگاشت بین فضای ورودی و فضای خروجی از یک مجموعه داده ورودی - خروجی، مهم ترین مساله در کاربردهای دنیای حقیقی می باشد. به جای بیان رابطه تابع بین ورودی و خروجی، تنها مجموعه جفت زوجی، از داده ورودی و خروجی فراهم می باشد. داده ورودی شامل  $N$  بردار با ابعاد  $m$  و در مقابل داده خروجی  $N$  مقدار حقیقی می باشد. ما به دنبال تابع ناشناخته ای هستیم که شرایط درون یابی را برآورده کند [۶].

$$x_i \in R^m, i = 1, 2, \dots, N \quad (2-20)$$

$$d_i \in R^1, i = 1, 2, \dots, N \quad (2-21)$$

$$f(x): R^m \rightarrow R^1 \quad (2-22)$$

$$f(x_i) = d_i, i = 1, 2, \dots, N \quad (2-23)$$

کیفیت تطبیق تابع تقریب زده شده و تابع اصلی، توسط یک تابع خطا تعیین می شود. تابعی که عموماً مورد استفاده قرار می گیرد طبق رابطه (۲-۲۴) بیان می شود.



$$E(f) = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2 = \frac{1}{N} \sum_{i=1}^N (d_i - f(x_i))^2 \quad (2-24)$$

به طور خلاصه، مساله عمده، کمینه کردن تابع خطا می باشد. به عبارت دیگر، بهبود بازده تقریب، هدف عمده تقریب توابع می باشد.

روش های بسیاری برای تقریب توابع منتشر شده اند که مانند ابزار های تقریب توابع هستند، و شبکه های عصبی مصنوعی نیز یکی از آن ها می باشد. با استفاده از یک شبکه عصبی در حل یک مساله، از ساختاری شبکه ای بهره می گیریم، به این مفهوم که، ارتباط میان لایه ها و نرون ها همانند یک شبکه می تواند معماری های مختلفی داشته باشد. عموماً تعیین ساختار یک شبکه عصبی خود مساله ای پیچیده است و به سادگی قابل حل نمی باشد. زمانی که تعداد لایه های قابل آموزش و واحد های پردازش بسیار کم باشند، شبکه توانایی آموزش مساله مورد نظر را ندارد. زمانی که تعداد لایه ها و نرون ها زیاد باشد، فرآیند آموزش بسیار طولانی خواهد بود. آموزش از روی نمونه ها بدین معنی است که شبکه با توجه به دانش حاصل از مجموعه نمونه ها توانایی یافتن تابع رابط بین ورودی و خروجی را دارا می باشد. این بدین معنی است که، پس از انجام گام آموزش، به ازای هر ورودی، با توجه به شرایط به دست آمده در گام آموزش مقدار خروجی متناظر با آن قابل پیش بینی خواهد بود. از این دیدگاه، آموزش مترادف با تقریب تابع می باشد [7].

یک شبکه عصبی مصنوعی ۳ لایه توانایی تقریب یک تابع غیر خطی را با هر دقت مورد نظری دارا می باشد. بنابراین جای تعجب نیست که، شبکه های عصبی مصنوعی دارای کاربردهای بیشماری می باشد، به ویژه با توجه به قابلیت شبکه، در یافتن الگو بین ورودی و خروجی، بدون نیاز به مدل تعیین شده از قبل. همچنین در زمینه های مرتبط با تقریب توابع نیز دارای کاربردهای بسیاری می باشد. در میان شبکه های عصبی، شبکه عصبی پرسپترون چند لایه، بیشتر مورد استفاده قرار می گیرد [8].

از سال ۱۹۸۶ میلادی، محبوب ترین شبکه عصبی، شبکه پرسپترون چند لایه می باشد، و محبوب ترین الگوریتم آموزش، روش پس انتشار خطا می باشد. با وجود این که، شبکه های کلاسیک پرسپترون چند لایه، دارای مزایای بسیاری می باشند، اما این مساله را نباید نادیده گرفت که دارای معایبی نیز می باشند. یکی از معایب بزرگ این شبکه ها، سرعت کم مرحله آموزش می باشد، که این امر به دلیل تعداد زیاد لایه های آموزش و نیاز به پس انتشار خطا می باشد، هرچند می توان با کاهش تعداد لایه ها سرعت آموزش را تا حدی افزایش داد. شبکه عصبی پرسپترون چند لایه به سرعت به جواب نمی رسد، در ابتدا به سمت گیر افتادن در یک کمینه محلی متمایل می شود، تا بتواند در یک فضای جستجوی بسیار پیچیده، به نقطه بهینه دست یابد. همچنین، آموزش یک شبکه عصبی پرسپترون چند لایه اصلی، با توجه به سرعت کم در همگرایی، بسیار زمان بر می باشد، از طرفی اگر میزان غیرخطی بودن بسیار زیاد باشد، شبکه عصبی پرسپترون چند لایه، همگرا نخواهد شد. این مسائل، بازده شبکه عصبی پرسپترون چند لایه را در تقریب توابع کاهش می دهد [۹].

در اینجا شبکه های عصبی توابع پایه شعاعی و شبکه های عصبی مبتنی بر موجک، برای تخمین توابع متناوب، نمایی و پیوسته تکه ای معرفی شده اند، که پاسخ مناسب تری نسبت به شبکه های پرسپترون دارند. انواع مختلفی از توابع پایه به عنوان توابع فعالسازی در لایه مخفی شبکه عصبی توابع پایه شعاعی و شبکه های عصبی مبتنی بر موجک به کار گرفته می شوند.

شبکه های عصبی توابع پایه شعاعی نیز نوع متداول دیگری از شبکه های عصبی می باشند، که در طرح مساله تقریب توابع در این پایان نامه مورد استفاده قرار گرفته است. برای غلبه بر مشکلات شبکه عصبی پرسپترون چند لایه، از شبکه های عصبی توابع پایه شعاعی استفاده می کنیم، که با جایگزینی توابع فعالسازی عمومی در شبکه پرسپترون، با توابع پایه شعاعی محلی، معرفی شده اند و پاسخ بهتری را نسبت به شبکه های عصبی پرسپترون در تقریب توابع نشان می دهند. توابع پایه شعاعی در ابتدا برای حل

مسائل درونیابی چند متغیره حقیقی معرفی شده اند. برومهد<sup>۱</sup> و لو<sup>۲</sup> اولین اشخاصی بودند که از توابع پایه شعاعی در طراحی شبکه های عصبی استفاده کردند [۱۰].

## ۲.۲.۲. شبکه های عصبی توابع پایه شعاعی

شبکه های عصبی توابع پایه شعاعی، شبکه های عصبی مسیر پیش خور می باشند، که شامل یک لایه ورودی، یک لایه میانی و یک لایه خروجی هستند. شبکه، یک نگاشت غیر خطی از فضای ورودی روی فضای مخفی میانی انجام می دهد و در ادامه یک نگاشت خطی از لایه مخفی میانی، روی فضای خروجی صورت می گیرد.

واحد های پردازشگر لایه مخفی در شبکه های عصبی توابع پایه شعاعی، با واحد های پردازشگر شبکه های پرسپترون چند لایه، متفاوت می باشند، توابع فعالسازی در شبکه های عصبی توابع پایه شعاعی، توابع پایه شعاعی (مانند توابع گاوسی) می باشند. این توابع عموماً دارای ۲ پارامتر مرکز و عرض می باشند.

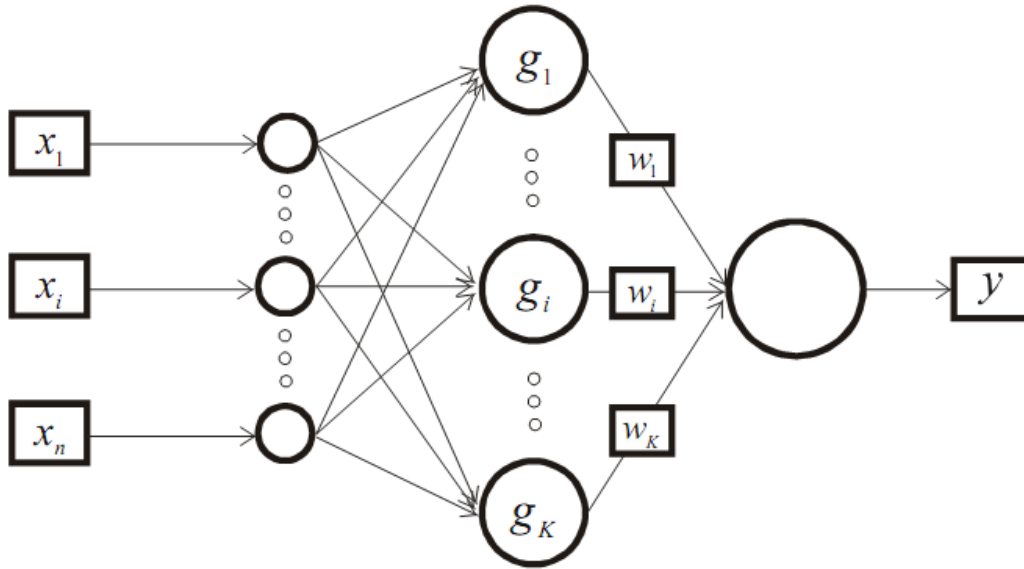
لایه خروجی نیز از واحدهای پردازشگر تشکیل شده است، که مقادیر نرمال، وزن دهی شده خطی تولید می کنند و هر واحد، یک خروجی تولید می کند. شبکه، ویژگی ساده ای دارد که در آن، وزن های بین ورودی و لایه مخفی، ۱ می باشد.

ساختار شبکه عصبی پایه توابع شعاعی در شکل (۲-۱) آورده شده است.

---

<sup>۱</sup> Bromhead

<sup>۲</sup> Lowe



شکل ۱-۲: ساختار شبکه عصبی توابع پایه شعاعی

در شکل (۱-۲)،  $x$  بردار ورودی،  $g$  تابع پایه شعاعی،  $w$  وزن های لایه خروجی و  $y$  خروجی شبکه می باشد. خروجی شبکه از رابطه (۲-۲۵) به دست می آید و  $c_i$  پارامتر مرکز توابع مربوط به نرون  $i$  می باشد.

$$y = \sum_{i=1}^k w_i g_i(x) = \sum_{i=1}^k w_i g(\|x - c_i\|) \quad (2-25)$$

توابع گاوسی عموماً با رابطه (۲-۲۶) بیان می شوند.

$$g_i(x) = e^{-\frac{\|x-c_i\|^2}{2\sigma_i^2}} \quad (2-26)$$

در رابطه (۲-۲۶)،  $\sigma_i$  پارامتر تغییر مقیاس تابع، متناظر با نرون  $i$  می باشد.

روش هایی برای انتخاب پارامترهای  $(c_i, \sigma_i)$  مربوط به توابع فعالسازی وجود دارد، اگر برخی نقاط آموزش در دسترس باشند، تمامی آن ها را می توان به عنوان پارامتر مرکز برای نرون های متناظر در نظر گرفت. در این حالت تعداد واحد های پردازشگر در لایه مخفی برابر با تعداد نقاط آموزش می باشد. اگر تعداد

نقاط آموزش زیاد باشد، آنگاه ممکن است استفاده از تمامی این نقاط به عنوان مراکز نرون ها از لحاظ حجم پردازش مناسب نباشد و لذا تمامی آن ها استفاده نشوند، در این حالت، داده های ورودی به گروه ها و دسته هایی تقسیم می شوند داده های مشابه در یک دسته قرار می گیرند و یکی از نقاط به عنوان مرکز دسته انتخاب می شود و یک نرون متناظر با این مرکز دسته، برای مجموعه داده های آموزش مشابه با هم در نظر گرفته می شود. این گروه های آموزشی مشابه را می توان توسط روش های خوشه یابی شناسایی کرد [۱۱].

### ۳.۲.۲. الگوریتم گرادیان نزولی برای آموزش شبکه های عصبی توابع پایه شعاعی

لایه مخفی شبکه های عصبی توابع پایه شعاعی را می توان توسط الگوریتم های آموزشی ناظر آموزش داد. برای این کار، یک الگوریتم مبتنی بر گرادیان نزولی را می توان در نظر گرفت، و هدف آموزش، تعیین وزن های هر نرون در شبکه می باشد.

اگر مجموعه نمونه های آموزشی را، مانند رابطه (۲-۲۷) در نظر بگیریم، یک الگوریتم خوشه یابی روی مجموعه نقاط در نظر گرفته می شود. مراکز خوشه ها  $c_i$ ، در نظر گرفته می شوند.

$$T = \{(x_i, z_i) | x_i \in R^n, z_i \in R, i = 1, 2, \dots, N\} \quad (2-27)$$

پارامترهای  $\sigma_i \in R, i = 1, \dots, K$ ، با توجه به قطر خوشه ها تعیین می شوند. این گام زمانی که  $K$  برابر با  $N$  باشد انجام نمی پذیرد، چون در این حالت،  $c_i = x_i, i = 1, \dots, N$  می باشد و هر نقطه ی آموزش یک مرکز خوشه می باشد و پارامتر های عرض برابر با  $\sigma_i = 1/N$  می باشد.

اگر توابع گاوسی به عنوان توابع فعالسازی در نظر گرفته شوند، سپس خطای آموزش مرحله  $l$ ، برابر با رابطه (۲-۲۸) خواهد بود.

$$E_l = \frac{1}{N} \cdot \sum_{i=1}^N (z_i - y_i)^2 \quad (28-2)$$

$$y_i = \sum_{j=1}^K w_j \cdot e^{-\frac{\|x-c_j\|^2}{2\sigma_j^2}}, i=1, \dots, K \quad (29-2)$$

$$\Delta w_i = -\eta \cdot \frac{\partial E}{\partial w_i}, i = 1, \dots, K \quad (30-2)$$

در رابطه (۳۰-۲)،  $\eta$  نرخ آموزش و  $E$ ، خطای مجموع آموزش می باشد. و اصلاح وزن ها با توجه به رابطه (۳۱-۲) انجام می شود.

$$w_i = w_i + \Delta w_i, i = 1, \dots, K \quad (31-2)$$

زمانی که فرآیند آموزش پایان یافت، نقاطی که از مجموعه آموزش نیستند، به طور تصادفی تولید می شوند. خطای تولید توسط رابطه (۳۲-۲) بیان می شود [۱۲].

$$E_g = \frac{1}{M} \cdot \sum_{i=1}^M (z_i - y_i)^2 \quad (32-2)$$

#### ۴.۲.۲. تقریب و درونیابی با استفاده از شبکه های عصبی توابع پایه شعاعی

مساله درونیابی، به صورت زیر تعریف می شود:

مجموعه ای از نقاط متفاوت به صورت رابطه (۳۳-۲)، و مجموعه ای از  $N$  عدد حقیقی طبق رابطه (۲-۲)

(۳۴) داده می شوند، حال باید تابعی را بیابیم که شرایط درونیابی را برآورده کند.

روش تابع پایه شعاعی، شامل انتخاب یک تابع به صورت رابطه (۳۳-۲) می باشد.

$$F(x) = \sum_{i=1}^N w_i g(\|x - x_i\|) \quad (33-2)$$

که در آن مجموعه (۳۴-۲) شامل  $N$ ، تابع پایه شعاعی می باشد. نقاط شناخته شده  $x_i \in \mathbb{R}^p, i = 1, \dots, N$ ، به مراکز توابع پایه شعاعی انتقال داده می شوند.

$$\{g(\|x - x_i\|) | i = 1, \dots, N\} \quad (34-2)$$

یک شبکه توابع پایه شعاعی با یک واحد پردازشگر در لایه خروجی در نظر گرفته می شود و  $N$  واحد در لایه مخفی در نظر گرفته می شوند، که  $\{g(\|x - x_i\|) | i = 1, \dots, N\}$ ، مجموعه ای از توابع فعالسازی برای واحد های پردازشگر مخفی می باشند. مساله درونیابی برای تعیین وزن ها کاهش می یابد.

ایده ترکیب توابع پایه موجک و شبکه های عصبی ۳ لایه، منجر به شبکه های عصبی مبتنی بر موجک شده است، که ساختاری مشابه با شبکه های عصبی توابع پایه شعاعی دارند. از زمانی که برای اولین بار شبکه های عصبی موجک معرفی شدند، با توجه به بهبود بزرگی که در مقابل شبکه های عصبی پرسپترون دارند تاکنون مورد علاقه بسیاری از پژوهشگران واقع شده اند [۱۳].

## ۵.۲.۲. شبکه های عصبی موجک

با در نظر گرفتن مشخصه محلی سازی زمان - فرکانس در توابع پایه موجک و توانایی آموزش شبکه های عصبی مصنوعی، شبکه های عصبی موجک تبدیل به ابزاری مفید و مناسب در تقریب توابع شده اند.

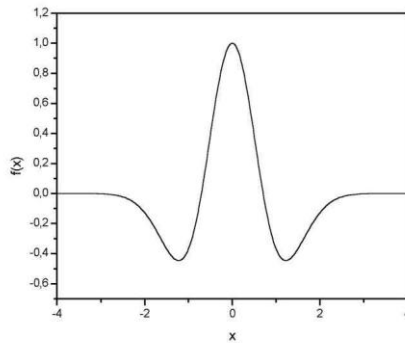
شبکه های عصبی موجک، معماری و ساختار مشابه با شبکه های عصبی توابع پایه شعاعی دارند با این تفاوت که به جای استفاده از توابع پایه شعاعی، از پنجره های موجک مانند موجک گاوسی، مورلت و کلاه مکزیکی در توابع فعالسازی در لایه مخفی استفاده می شود، و پارامترهای بردار مراکز و عرض، به ترتیب با بردارهای انتقال و مقیاس تعویض می شوند.

مشابه با شبکه های عصبی توابع پایه شعاعی، شبکه های عصبی موجک مبتنی بر فاصله هذلولوی بین بردار ورودی و بردار انتقال هستند و سپس هر یک از اجزای فاصله توسط بردار مقیاس وزن دهی می شود. بنابراین خروجی شبکه عصبی موجک از رابطه (۲-۳۵) به دست می آید.

$$y = \sum_{j=1}^M w_j \Phi_j \left( \left\| \frac{\vec{X} - \vec{E}_j}{\vec{T}_j} \right\| \right) \quad (۲-۳۵)$$

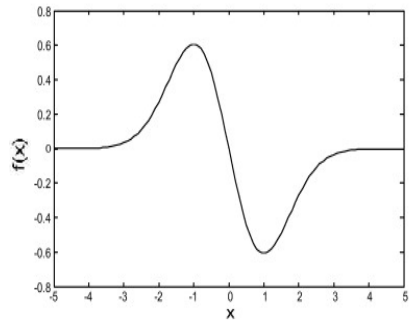
که در آن،  $M$  تعداد نقطه های مخفی می باشد.

آموزش شبکه عصبی موجک مشابه با شبکه عصبی توابع پایه شعاعی می باشد، که در آن پارامترهای مرکز و عرض توابع پایه شعاعی با پارامترهای انتقال و مقیاس توابع پایه موجک جایگزین شده اند [۱۳].

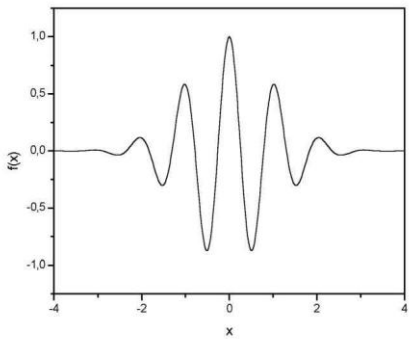


شکل ۲-۲: موجک کلاه مکزیکی





شکل ۲-۳: موجک گاوسی



شکل ۲-۴: موجک مورلت

۳. الگوریتم بهینه سازی توده ذرات

### ۱.۳. معرفی

از زمانی که الگوریتم های محاسباتی متداول به دلیل ساختارهای غیر انعطاف پذیر که به دلیل نویری بودن داده ها، ناکامل به نظر رسیدند و در مسایل چند بعدی، توانایی حل کردن مسایل دنیای واقعی را نداشتند، نظریه های محاسبات طبیعی، برای حل این گونه مسایل جایگزین شوند. این نظریه ها شامل عناصر ساده ای می باشند که زمانی که با هم در نظر گرفته می شوند، توانایی حل مسایل پیچیده دنیای واقعی را دارند. در حالت عام، نظریه های محاسباتی طبیعی به ۳ دسته تقسیم می شوند [۱۴]:

۱. پسزایی<sup>۱</sup>

۲. تکامل نژادی<sup>۲</sup>

۳. هستی زایی<sup>۳</sup>

گروه پسزایی شامل مسائلی است که در آن باید، ساختار پیچیده ای را بهبود داد، برای این کار لازم است تا یک آموزش موقت انجام داد. یک مثال واضح از این گروه شبکه های عصبی مصنوعی می باشند، که در آن سیستم پیچیده مغز انسان شبیه سازی می شود.

گروه تکامل نژادی مربوط به الگوریتم های تکاملی می باشد. در الگوریتم های وابسته به این گروه، رقابت برای حفظ بهترین تطابق می باشد. الگوریتم های وابسته به این گروه شامل برنامه نویسی تکاملی<sup>۴</sup>، برنامه نویسی ژنتیک<sup>۵</sup> و تکامل تفاضلی<sup>۱</sup> می باشد.

---

<sup>۱</sup> Epigenesis

<sup>۲</sup> phylogeny

<sup>۳</sup> Ontogeny

<sup>۴</sup> Evolutionary Programming (EP)

<sup>۵</sup> Genetic Programming (GP)

گروه هستی زایی شامل الگوریتم های می باشد که رفتارهای خاصی از طبیعت را که در آن اتفاق می افتد می پذیرد. الگوریتم هایی مانند بهینه سازی توده ذرات و الگوریتم ژنتیک از این دسته می باشند و در حقیقت، در مقایسه با سایر الگوریتم ها یک طبیعت مبتنی بر همکاری دارند. مزیت روش های ذکر شده، در توانایی آن ها در بهبود پاسخ در کاربردهای مختلف و عدم نیاز به دانش قبلی نسبت به فضای جستجو می باشد. روش های ذکر شده قبلی، تضمینی در یافتن پاسخ بهینه مساله ارائه نمی دهند و در آن ها، برای رسیدن به تابع تطبیق، نیاز به تعداد محاسبات زیاد در تکرارهای زیاد می باشد. در میان نظریه های بیان شده، از آنجا که الگوریتم بهینه سازی توده ذرات در عین سادگی، طبیعت موثرتری دارد برای مطالعات و تحقیقات مناسب تر به نظر می رسد. همچنین می تواند به عنوان پایه ای برای الگوریتم های تکاملی مهم، به کار گرفته شود.

مهم ترین شباهت بین این نظریه ها و الگوریتم ژنتیک، دارا بودن جمعیت هم کنش می باشد. این الگوریتم ها در مقایسه با الگوریتم ژنتیک سرعت بالاتری در یافتن پاسخ هایی نزدیک به نقطه بهینه دارند و در همگرایی زودرس، سریعتر از الگوریتم ژنتیک می باشند.

الگوریتم بهینه سازی توده ذرات، یک روش بهینه سازی مبتنی بر جمعیت می باشد که بر اساس رفتار اجتماعی پرواز پرندگان یا دسته ماهی ها در سال ۱۹۹۵ توسط دکتر کندی<sup>۲</sup> و دکتر ابرهارت<sup>۳</sup> معرفی گردید [۱۵].

الگوریتم بهینه سازی توده ذرات، شباهت های بسیاری با روش های محاسباتی تکاملی، مانند الگوریتم ژنتیک دارد، سیستم در ابتدا با مجموعه پاسخ های تصادفی، مقداردهی می شود و با به روزرسانی مرحله

---

<sup>۱</sup> Differential Evolutionary (DE)

<sup>۲</sup> Kennedy

<sup>۳</sup> Eberhart

به مرحله، به دنبال نقطه بهینه می گردد. هرچند بر خلاف الگوریتم ژنتیک، الگوریتم بهینه سازی توده ذرات دارای عملگرهای تکاملی مانند جهش و نقاط همگذر نمی باشد. در الگوریتم بهینه سازی توده ذرات پاسخ های احتمالی، "ذرات" نامیده می شوند، که در فضای مساله به دنبال نقطه بهینه به پرواز در آمده اند. در مقایسه با الگوریتم ژنتیک، الگوریتم بهینه سازی توده ذرات دارای مزایایی می باشد، از جمله این که به سادگی قابل پیاده سازی می باشد و پارامترهای کمی برای بهینه سازی دارد.

همانگونه که بیان شد، الگوریتم بهینه سازی توده ذرات، رفتار پرواز پرندگان را شبیه سازی می کند. این رفتار این گونه تعریف می شود: یک گروه از پرندگان در یک فضا به طور تصادفی به دنبال غذا می گردند، در فضای جستجو یک تکه غذا وجود دارد، تمامی پرندگان مکان غذا را نمی دانند، اما آنها می دانند در هر تکرار چه مقدار از محل مورد نظر فاصله دارند، در این حالت بهترین استراتژی برای یافتن محل غذا این است که نزدیک ترین پرنده به محل مورد نظر را تعقیب کنیم.

الگوریتم بهینه سازی توده ذرات مانند بیان بالا عمل می کند و از آن برای مسائل بهینه سازی استفاده می کند. در الگوریتم بهینه سازی توده ذرات، هر پاسخ یک پرنده در فضای جستجو می باشد، که آن را ذره می نامیم. همه ی ذرات دارای یک مقدار هدف می باشند که این مقدار توسط یک تابع هدف تعیین می شود تا بهینه شود و دارای سرعت می باشند که پرواز ذرات را هدایت می کند.

الگوریتم بهینه سازی توده ذرات، با یک گروه تصادفی از ذرات (پاسخ) مقداردهی اولیه می شود، سپس با به روز رسانی مرحله به مرحله، مقدار بهینه را جستجو می کند. در هر تکرار، هر ذره با توجه به دو مقدار برتر به روزرسانی می شود. اولین مقدار بهترین پاسخی است که تا کنون به دست آمده و "بهترین پاسخ شخصی" نامیده می شود و دیگری بهترین پاسخ به دست آمده توسط هر ذره در جمعیت می باشد و "بهترین پاسخ جمعیت یا عمومی" نامیده می شود.

پس از یافتن این دو مقدار برتر، مکان جدید ذرات با توجه به روابط (۱-۳)، (۲-۳) به روزرسانی می شود.

$$x^i(k+1) = x^i(k) + v^i(k+1) \quad (1-3)$$

$$v^i(k+1) = wv^i(k) + c_1r_1(x_{pbest}^i - x^i(k)) + c_2r_2(x_{gbest}^i - x^i(k)) \quad (2-3)$$

در روابط ذکر شده، پارامترهای استفاده شده به صورت زیر تعریف می شوند:

$x^i(k)$ : موقعیت ذره  $i$  در لحظه  $k$

$v^i(k)$ : سرعت حرکت ذره  $i$  به سمت هدف در لحظه  $k$

$w$ : ضریب اینرسی وزن

$c_1$  و  $c_2$ : ضرایب شتاب دهنده

$r_1$  و  $r_2$ : مقادیر تصادفی، بین ۰ و ۱

$x_{pbest}^i$ : بهترین مکان یافت شده برای ذره  $i$  (بهترین پاسخ شخصی)

$x_{gbest}$ : بهترین مکان یافت شده تا کنون (بهترین پاسخ عمومی)

شبه کد پروسه بهینه سازی توده ذرات در زیر آورده شده است:

- مقداردهی اولیه پارامترهای بهینه سازی توده ذرات
- برای تعداد تکرارهای الگوریتم بهینه سازی توده ذرات
  - به روز رسانی موقعیت توده ذرات با توجه به موقعیت قبلی و سرعت
  - تولید تابع هدف و محاسبه مقدار خطا
  - به روز رسانی مقادیر بهترین پاسخ شخصی و بهترین پاسخ عمومی

○ به روز رسانی روابط سرعت

○ پایان

• پایان

در بهینه سازی توده ذرات، گروه به صورت نمونه، به وسیله ذرات در فضای چند بعدی که مکان و سرعت دارد، مدل سازی می شود .

این ذرات در میان این ابر فضا (فضای دارای بیش از سه بعد)، پرواز می کنند و دو توانایی ضروری دارند:

۱- حافظه ای برای ذخیره سازی بهترین مکان خود

۲- آگاهی در مورد بهترین موقعیت در همسایگی خود یا در کل فضای پاسخ ها. ذرات، مکان های

خوب را به یکدیگر از طریق ارتباط انتقال می دهند و موقعیت و سرعتشان را با مکان های خوب

تنظیم می کنند .

هر ذره برای اعمال تغییری مناسب در مکان و سرعت خود اطلاعات زیر را دارا می باشد :

۱- "بهترین پاسخ عمومی" که برای همه شناخته شده است و هنگامی که هر ذره بهترین مکان

جدیدی را شناسایی کند، فوراً برای بقیه ذرات اطلاعات مربوطه را به روز رسانی می کند .

۲- "بهترین پاسخ همسایگی" که ذره از طریق ارتباط با زیر مجموعه های گروه، آن را بدست می

آورد .

۳- "بهترین پاسخ شخصی" که بهترین راه حلی است که ذره تا کنون تجربه کرده است .

همه ذرات شروع به تاثیر پذیری از "بهترین پاسخ عمومی" می کنند تا سرانجام به آن نزدیک شوند. ذرات

در فضای جستجو در نزدیکی "بهترین پاسخ عمومی" حرکت می کنند و بقیه فضا را کاوش نمی کنند، به

این پدیده "همگرایی" گفته می شود. اگر ضریب اینرسی سرعت را کوچک انتخاب کنیم، تمام ذرات می توانند سرعتشان را کاهش دهند تا اینکه در "بهترین پاسخ عمومی" به سرعت صفر نزدیکتر شوند. یک راه خروج از وضعیت همگرایی اولیه (همگرایی نامطلوب) این است که دوباره به موقعیت ذرات (پس از رخ دادن همگرایی) مقدار اولیه بدهیم [۱۶].

### ۲.۳. نسخه اصلی بهینه سازی توده ذرات

ضرایب شتاب دهنده  $C_1$  و  $C_2$ ، بیانگر شتابی هستند که به هر ذره داده می شوند تا به سمت محل بهترین پاسخ شخصی و بهترین پاسخ عمومی هدایت شوند. تنظیم این ضرایب، مقدار تنش در سیستم را تغییر می دهد. مقدار کم برای این ضرایب باعث می شود که ذرات قبل از کشیده شدن به سمت پاسخ به آرامی حرکت کنند، در حالی که مقدار زیاد، منجر به حرکت سریع یا عبور از منطقه هدف می شود.

تجربه به دست آمده از اجرای الگوریتم، نشان می دهد که عموماً می توان در اکثر موارد، مقدار این ثوابت را برابر با یکدیگر و برابر با ۲۰ در نظر گرفت. بیشینه ی سرعت، متغیری است که عموماً، ما آن را به صورت ثابت، در حدود ۱۰ تا ۲۰ درصد از محدوده دینامیک متغیر در هر بعد، در نظر می گیریم.

می توان بر اساس شبیه سازی های اجتماعی، نسخه محلی از بهینه سازی توده ذراتی را در نظر گرفت، در این نسخه، ذرات شامل اطلاعاتی از بهترین پاسخ های خود و همسایگی خود، به جای کل مجموعه هستند. به عنوان مثال اگر اندازه همسایگی ۲ در نظر گرفته شود ذره ی  $i$  با ذره ی  $(i - 1)$  و ذره ی  $(i + 1)$  مقایسه می شود. همسایه ها، به صورت همسایه مکانی تعریف می شوند و همسایه ها و همسایگی ها در طول اجرای الگوریتم تغییر نمی کنند. برای نسخه همسایگی، تنها تغییر در پروسه شامل ۶ گام جایگزینی بهترین پاسخ محلی به جای بهترین پاسخ عمومی می باشد. تجربه نشان داده اندازه



همسایگی عموماً ۱۵ درصد از اندازه جمعیت را می توان برای بسیاری از کاربردها در نظر گرفت. انتخاب اندازه جمعیت بستگی به تعریف مساله دارد. عموماً اندازه ۲۰ تا ۵۰ برای مسائل انتخاب مناسبی می باشد.

### ۱.۲.۳. وزن اینرسی

بیشینه سرعت، برای کنترل توانایی اکتشاف یک توده ذره به کار می رود. همانگونه که پیش تر بیان شد بیشینه سرعت زیاد، اکتشاف عمومی را ساده می کند در حالی که بیشینه سرعت کمتر، اکتشاف محلی را بهبود می دهد. تعریف اینرسی وزن، به منظور کنترل بهتر اکتشاف و بهره برداری معرفی گردید. محرک، این قابلیت را دارد که نیاز به بیشینه سرعت را از بین ببرد. نیاز به اینرسی وزن برای اولین بار در مقاله ای در سال ۱۹۹۸ توسط ابرهارت و شای<sup>۱</sup> گزارش شد [۱۶].

استفاده از اینرسی وزن، باعث بهبود کارایی در برخی کاربردها گردیده است. اینرسی وزن، عموماً در طول پروسه با مقدار ۰.۹ شروع شده و به صورت خطی کاهش یافته و به مقدار ۰.۴ ختم می شود. انتخاب مناسب اینرسی وزن، توازنی بین اکتشاف و بهره برداری عمومی و محلی برقرار می کند و با تعداد تکرار کمتر به یک پاسخ مناسب می رسد.

پس از تجربیاتی در مورد اینرسی وزن، به این نتیجه دست یافته شد، که عامل بیشینه سرعت همواره نمی تواند حذف شود، الگوریتم بهینه سازی توده ذرات، زمانی به خوبی عمل می کند که بیشینه سرعت متناسب با محدوده دینامیک هر متغیر (هر بعد) انتخاب شود. بنابراین نیاز به تصمیم گیری در مورد چگونگی انتخاب بیشینه سرعت نمی باشد.

---

<sup>۱</sup> Shai

### ۲.۲.۳. ضریب تراکم

در کاری که توسط کلرک<sup>۱</sup> در سال ۱۹۹۹ صورت گرفت بیان شد که استفاده از یک ضریب متراکم کننده ممکن است برای همگرایی الگوریتم بهینه سازی توده ذرات مورد نیاز باشد. این ضریب متراکم کننده در واقع به صورت تابعی از ضرایب شتاب دهنده می باشند. از آنجا که تصور می شد پس از استفاده از ضریب متراکم کننده نیازی به استفاده از بیشینه سرعت نمی باشد، پس در آزمایشات و کاربردهای اولیه بیشینه سرعت مقدار ۱۰۰۰۰۰ انتخاب می شد. هرچند در آزمایشات و کاربردهای بعدی طبق یک قانون سرانگشتی بیشینه سرعت را به بیشینه مقدار هر بعد محدود می کردند [۱۷].

### ۳.۲.۳. کاربردها

یکی از دلایلی که بهینه سازی توده ذرات بسیار پرطرفدار می باشد، به این خاطر است که تعداد پارامترهای بسیار کمی برای تنظیم دارد. نسخه ای که تغییرات بسیار کمی دارد (یا کلاً بدون تغییرات) روی محدوده بسیار وسیعی از کاربردها به خوبی عمل می کند.

بهینه سازی توده ذرات برای دستیابی به پاسخ مناسب در محدوده وسیعی از کاربردها استفاده می شود و در همه کاربردهای خاص برای نیازهای خاص، می تواند به خوبی مورد استفاده واقع شود.

اولین کاربرد، روشی را معرفی می کند که می تواند در بسیاری از موارد، مورد استفاده قرار گیرد و شامل شبکه های عصبی مصنوعی می باشد. بهینه سازی توده ذرات تنها وزن های شبکه را شامل نمی شود بلکه ساختار شبکه را نیز شامل می شود. این روش بسیار ساده و موثر می باشد و ما عمدتاً از آن به صورت کامل برای آموزش شبکه به صورت انتشار بازخوردی استفاده می کنیم [۱۸].

---

<sup>۱</sup> Clerk

به عنوان مثالی از اعمال روی شبکه عصبی، بهینه سازی توده ذرات روی تحلیل لرزش های انسان استفاده شد. تشخیص لرزش های بدن، شامل بیماری پارکینسون و لرزش های خاص در مناطق چالش برانگیز می باشد. بهینه سازی توده ذرات، برای آموزش شبکه عصبی به منظور تشخیص، بین عوامل عادی و لرزش ها استفاده می شود. ورودی شبکه، دامنه های نرمال شده به دست آمده از سیستم حرکت نگار می باشد، این روش سریع و دقیق می باشد [۱۹].

به عنوان مثالی دیگر از کاربرد شامل شبکه عصبی، می توان آسیاب کردن نهایی را نام برد که یکی از عمومی ترین و اساسی ترین مراحل در حذف فلز از محیط می باشد. با پیشرفت علوم کامپیوتر و تولید ابزارهای ماشینی، کنترل عددی به طرز موثری بهبود داده شد، با این حال هنوز هم از نقطه بهینه فاصله زیادی وجود داشت و هیچ کدام از روش های قبلی نیز به اندازه کافی برای پیاده سازی با دقت بالا کافی نبودند. یک کاربرد جدید و موفق، با استفاده از شبکه عصبی و بهینه سازی توده ذرات برای شبیه سازی پروسه، برای بهینه سازی چند بعدی معرفی شد و مورد استفاده واقع گردید، این کار توسط ابزار طراحی با استفاده از کامپیوتر و تولید به کمک کامپیوتر<sup>۱</sup> و دیگر ابزارهای مهندسی استاندارد پیاده سازی شده است [۲۰].

کاربرد دیگری از بهینه سازی توده ذرات، استفاده از آن برای کنترل ولتاژ و توان راکتیو مورد نیاز در شبکه برق ژاپن می باشد. در اینجا، بهینه سازی توده ذرات، برای تعیین استراتژی کنترلی با توجه به متغیرهای کنترلی پیوسته و گسسته می باشد، که منتج به یک الگوریتم مختلط، با مقادیر دودوئی و حقیقی می شود. تثبیت ولتاژ در سیستم، توسط یک روش جریان توان مستقیم، به دست آمد [۲۱].

---

<sup>۱</sup> CAD/CAM

به عنوان کاربردی دیگر، بهینه سازی توده ذرات با الگوریتم پس انتشار، برای آموزش شبکه عصبی به منظور تخمینگر میزان شارژ باطری، در وسایل نقلیه الکتریکی مورد استفاده قرار می گیرد. تعیین میزان شارژ، مساله ای مهم در تکنولوژی وسایل نقلیه الکتریکی یا دوگانه می باشد. میزان شارژ در واقع همان میزان سوخت یک وسیله نقلیه الکتریکی می باشد. یک روش مورد استفاده، ترکیب بهینه سازی توده ذرات و الگوریتم پس انتشار می باشد، ابتکار در استفاده از این روش، برای بهینه سازی مجموعه ای از داده ها می باشد، نتایج حاصل برتری این روش را اثبات می کند [۲۲].

در نهایت یکی از کاربردهای جالب در این مورد، توسط یک شرکت آمریکایی، به منظور بهینه سازی ترکیب عناصر<sup>۱</sup> ارائه شده است. در این کار ترکیب عناصر، به ترکیب عناصر، به منظور افزایش تولید میکرو ارگانیزم ها که به صورت طبیعی، غیر قابل دستیابی می باشند یا تولید آن ها، اشاره دارد. در اینجا بهینه سازی توده ذرات، به صورت موازی با روش های بهینه سازی صنعتی، در فضای عناصر در قسمت های مختلف استفاده شده است. بهینه سازی توده ذرات پایدار نشان داده است که مقدار یک عنصر ممکن است در تکرارهایی در جستجو خلل ایجاد کند، اما همواره باعث می شود پاسخی نهایی نتایج مطلوبی داشته باشد. طبیعت بهینه سازی توده ذرات، نسبت های بیشتری از فضای مساله را در مقایسه با سایر روش های عام بر می گیرد [۱۸].

### ۳.۳. دسته بندی بهینه سازی توده ذرات

در این بخش، دسته بندی های متفاوت از بهینه سازی توده ذرات را بیان می کنیم [۲۳].

---

<sup>۱</sup> Ingredient Mix

### ۱.۳.۳. پیوستگی

از نقطه نظر پیوستگی، در فضایی که ذرات در آن قرار گرفته اند، بهینه سازی توده ذرات به دو دسته تقسیم می شود، بهینه سازی توده ذرات در فضای گسسته و بهینه سازی توده ذرات در فضای پیوسته. در حالت پیوسته، مسیر حرکت ذرات مشابه یک تغییر، در موقعیت یک ذره، در همان جهت می باشد. هرچند، در فضای گسسته، این مسیر حرکت مانند یک تغییر در احتمال مقدار موقعیت است که می تواند صفر یا یک باشد.

### ۲.۳.۳. مطابقت

گاهی اوقات، در طول اجرای بهینه سازی توده ذرات، فرآیند تکامل توده تقریباً متوقف می شود و ساکن می ماند. این امر عمدتاً به این دلیل است که برخی ذرات غیرفعال می شوند، بدین ترتیب، این ذرات، در جستجوی محلی و عمومی، ناتوان می شوند. بنابراین، موقعیت آن ها نسبت به قبل تغییر چندانی نمی کند و سرعت آن ها نزدیک به صفر خواهد بود. یک روش، جایگزینی فعال این ذرات با ذرات تازه می باشد به نحوی که چرخش های موجود در میان ذرات، که مبتنی بر ذرات بهینه سازی توده ذرات می باشند، حفظ شوند. این کار، با استفاده از روش بهینه سازی توده ذرات فعال انجام می گیرد. اما گاهی این توقف به دلیل تمایل توده برای رسیدن به حالت متعادل می باشد تا از جستجوی فضاهای بیشتر جلوگیری کند و ممکن است در بهینه محلی گیر کرده باشد. به منظور غلبه بر این مشکل، الگوریتم بهینه سازی توده ذرات پراکنده ساز مورد استفاده قرار می گیرد، یک سیستم پراکنده ساز، بی نظمی منفی ایجاد می کند که این کار، با ایجاد آشوب در میان ذرات صورت می گیرد. با استفاده از این سیستم، در عمل از حالت سکون ذکر شده، جلوگیری می شود. در اینجا، دو روش ذکر شده قبلی به طور همزمان مورد استفاده قرار می گیرند. در این حالت، از طرفی یک

پراکندگی برای بهینه سازی توده ذرات، بهبود داده می شود تا یک بی نظمی منفی، به سیستم معرفی کند، از طرفی دیگر، یک عملگر تحرک، برای زمانی که الگوریتم به یک حالت ساکن می رسد در نظر گرفته می شود، تا باعث افزایش تغییرات در توده شود. بنابراین، یک روش سازگار برای به روز رسانی اینرسی وزن در نظر گرفته می شود تا توازن بین بهینه سازی محلی و عمومی را حفظ کند. این کار توسط الگوریتم بهینه سازی توده ذرات با پراکنده سازی سازگار متغیر، صورت می گیرد. به وضوح، ایده های مورد نظر در هر دو محیط متغیر یا ساکن قابل دستیابی خواهند بود.

### ۳.۳.۳. پیکربندی

الگوریتم بهینه سازی توده ذرات، از نقطه نظر دستیابی به اطلاعات ذرات به پیکربندیهای مختلف تقسیم می شود. در نوع بهترین پاسخ عمومی، همه ذرات با یکدیگر مرتبط می باشند. در حقیقت، همه ذرات از یکدیگر تاثیر می گیرند. اما در وضعیت بهترین پاسخ محلی، هر ذره با ذرات همسایگی در ارتباط است و شبکه حلقه ای شکل می گیرد. پیکربندی دیگر، هرمی می باشد که همانند مثلث، سه بعدی است که ارتباط هر ذره را در سه بعد نشان می دهد. در پیکربندی ستاره ای، یک نقطه مرکزی، از همه جمعیت تاثیر می گیرد و همچنین بر همه آن ها تاثیر می گذارد. وضعیت کوچک، گرافی است که از ذرات و زیر گروه های عایق شده تشکیل شده است و در حقیقت مثالی از شبکه ناهمگون می باشد. در وضعیت ون-نیومن<sup>۱</sup>، همسایگی های بالا، پایین و هر طرف در یک فضای حلقه دو بعدی قرار می گیرد. در وضعیت بهترین-قدرت، در حقیقت حالت میانگینی از وضعیت بهترین پاسخ عمومی و بهترین پاسخ محلی ارائه می شود، در این وضعیت، اطلاعات نه تنها در بین ذراتی که در حالت گسسته در آن لحظه، در همسایگی قرار دارند تقسیم می شوند، همچنین در میان، همه ذراتی که در

---

<sup>۱</sup> Von - neumman

محدوده دید یک ذره قرار دارند نیز تقسیم می شوند. همچنین، ذراتی که در محدوده دید یکدیگر قرار دارند، از بهترین موقعیت یکدیگر آگاهی دارند و این امر می تواند شرایط را به حقیقت نزدیک تر گرداند. علاوه بر وضعیت های ذکر شده، وضعیت های دیگری نیز وجود دارند که به صورت تصادفی ایجاد می شوند.

### ۴.۳.۳. محدودیت ها

از نقطه نظر محدودیت، بهینه سازی توده ذرات به دو دسته محدود و غیر محدود تقسیم می شود. در حالت عمومی، که همان حالت کلاسیک الگوریتم می باشد، الگوریتم محدود می باشد. شایان ذکر است که در هر دو حالت، معادلات به روزرسانی سرعت و موقعیت، یکسان هستند و تفاوت در این است که در حالت کلاسیک، شرایط و کران های بالا و پایین برای سرعت و موقعیت تعریف می شوند.

### ۵.۳.۳. ترکیب با روش های دیگر ابتکاری

به منظور افزایش بازده الگوریتم، برای غلبه بر مشکلاتی مانند گیر کردن در بهینه محلی و همچنین افزایش واگرایی، به منظور بهتر یافتن پاسخ در بهینه سازی توده ذرات، این الگوریتم با روش های دیگر ابتکاری مانند تبرید شبیه سازی شده، کلونی مورچگان و الگوریتم ژنتیک ترکیب می شود.

## ۴.۳. الگوریتم های بهبود یافته بر پایه بهینه سازی توده ذرات

### ۱.۴.۳. بهینه سازی توده ذرات با هدف وفقی<sup>۱</sup>

در این الگوریتم علاوه بر دو عبارت موجود، با نام های "بهترین موقعیت" و "بهترین موقعیت قبلی" در به روزرسانی سرعت ذره، یک عبارت سوم با نام "هدف فعال" نیز استفاده می شود. محاسبه "هدف فعال" پیچیده می باشد و به موقعیت فعلی ذرات تعلق ندارد. این روش، تضمین می کند که همگرایی بهینه سازی توده ذرات به خوبی صورت گرفته و الگوریتم در بهینه محلی گیر نکرده است.

[۲۴].

### ۲.۴.۳. بهینه سازی توده ذرات فعال<sup>۲</sup>

در طول اجرای پروسه بهینه سازی توده ذرات، گاهی تعدادی از ذرات غیرفعال هستند، به همین دلیل است که آن ها توانایی جستجوی محلی و عمومی را ندارند و موقعیت آن ها تغییر نمی کند، بنابراین سرعت آن ها به نزدیکی صفر رسیده است. یک روش برای غلبه بر این مساله، جایگزینی این ذرات غیرفعال، با ذراتی تازه می باشد، به نحوی که روابط مبتنی بر بهینه سازی توده ذرات حفظ شوند.

[۲۵].

---

<sup>۱</sup> Active target PSO (APSO)

<sup>۲</sup> Adaptive PSO (APSO)



### ۳.۴.۳. بهینه سازی توده ذرات با پیش سازی زاویه<sup>۱</sup>

این روش از یک تابع مثلثاتی استفاده می کند تا یک رشته بیت تولید کند. تفاوت آن با الگوریتم بهینه سازی توده ذرات باینری، در بازده بالای محاسباتی آن است. به همین دلیل است که در آن، از تولید یک بردار باینری با ابعاد زیاد جلوگیری می شود و بنابراین پروسه گسسته آن پیچیده نمی باشد. هرچند در نهایت، این روش تمامی مسایل با ابعاد بالا را به مسئله ای با ۴ بعد تبدیل می کند. این روش مقدار زیادی از حافظه را حفظ می کند و اجرای آن آسان است [۲۶].

### ۴.۴.۳. الگوریتم بهینه سازی توده ذرات با بهترین چرخش<sup>۲</sup>

این الگوریتم برای بهینه سازی توابع چند حالته استفاده می شود و در حقیقت، توده گروه به چند زیر گروه تقسیم می شود. تقسیم و جدا کردن توده به چند زیرمجموعه، برای مساله تک حالته مطلوب نمی باشد. هرچند در بهینه سازی توده ذرات ساده، برای یک تابع چند حالته، احاطه گسترده نسبت به بازده کل توده، باعث همگرایی سریعتر سیستم می شود و همچنین احتمال گیر کردن در یک بهینه محلی را افزایش می دهد، اما در الگوریتم بهینه سازی توده ذرات با بهترین چرخش، زمانی که بهترین چرخش انجام شود، با اعمال نیرو به توده برای جابجایی از یک بهینه محلی به دیگر، از گیر کردن در بهینه محلی جلوگیری می شود و قدرت جستجوی بهینه در فضای جستجو را افزایش می دهد. در این الگوریتم، چرخش های متناوب، در میان ذرات زیر گروه های مختلف انجام می گیرد [۲۷].

---

<sup>۱</sup> Angle Modulated PSO (AMPSO)

<sup>۲</sup> Best Rotation PSO (BRPSO)

### ۵.۴.۳. بهینه سازی توده ذرات باینری<sup>۱</sup>

تفاوت بین بهینه سازی توده ذرات و بهینه سازی توده ذرات باینری بر اساس نواحی جستجوی تعریف شده می باشد. در بهینه سازی توده ذرات ساده، جابجایی در فضای جستجو به معنی تغییر در مقدار موقعیت در یک یا چند بعد می باشد. در حالیکه در بهینه سازی توده ذرات باینری، جابجایی در فضا، به معنی تغییر در احتمال مقدار موقعیت می باشد که دارای ۲ مقدار صفر و یک می باشد [۲۸].

### ۶.۴.۳. بهینه سازی توده ذرات شبه تکاملی<sup>۲</sup>

این الگوریتم در سال ۲۰۰۲ ارائه گردید، این روش، برای حل مسائل پیوسته به کار گرفته شده است. در این الگوریتم یک تبدیل لاگرانژیان معرفی می گردد که با استفاده از آن مسئله به یک مسئله کمینه - بیشینه با نقطه زینی شکل تبدیل می شود. سپس یک الگوریتم بهینه سازی توده ذرات شبه تکاملی، بر آن اعمال می شود. دو الگوریتم بهینه سازی توده ذرات، توسط تابع تطابق با هم مرتبط می شوند. یک الگوریتم بهینه سازی توده ذرات، در محاسبه مقدار تطبیق استفاده می شود و دیگری محیط را مهیا می کند [۲۹].

### ۷.۴.۳. بهینه سازی مقید با بهینه سازی توده ذرات<sup>۳</sup>

الگوریتم این روش برای مسائل مقید به یک هدف، قابل پیاده سازی می باشند. در این الگوریتم، یک روش برای تخمین قیود و فایل های خارجی که "سطح مدارا"<sup>۴</sup> نامیده می شوند، به کار گرفته می

---

<sup>۱</sup> Binary PSO (BPSO)

<sup>۲</sup> Co-evolutionary PSO (PSO)

<sup>۳</sup> Constrained optimization via PSO (COPSO)

<sup>۴</sup> Tolerant

شود، تا ذرات حفظ شوند. علاوه بر این، در این روش، برخی از ذرات در بین تنظیمات قبود از دست می روند. به منظور بهبود طول عمر این ذرات، فایل های خارجی ذکر شده در بالا برای طرح یک شبکه حلقه ای به کار گرفته می شود. در حقیقت، "بهینه سازی مقید با بهینه سازی توده ذرات"، ایجاد بهبود در نسخه ای از بهینه سازی توده ذرات می باشد، که شامل بهترین پاسخ محلی است [۳۰]، [۳۱].

### ۸.۴.۳. بهینه سازی توده ذرات پراکنده ساز<sup>۱</sup>

گاهی اوقات ارزشیابی در بهینه سازی توده ذرات، ساکن می شود که این امر به دلیل تمایل توده برای رسیدن به حالت تعادل می باشد. بنابراین الگوریتم از جستجو برای مناطق بیشتر جلوگیری می کند و به طور تصادفی ممکن است در بهینه محلی گیر کند. برای غلبه بر این مساله، یک سیستم پراکنده ساز ارائه می شود که بی نظمی منفی را به سیستم معرفی می کند و آشفتگی در ذرات تولید می کند [۳۲].

### ۹.۴.۳. بهینه سازی توده ذرات با محدوده تقسیم شده<sup>۲</sup>

در این روش چندین تابع هدف وجود دارد، با توجه به مقدار یکی از توابع، ذرات در ابتدا به زیر گروه هایی تقسیم می شوند. سپس الگوریتم گسسته بهینه سازی توده ذرات، در هر یک از زیر گروه ها اجرا می شود. اگر شرایط توقف برقرار باشد، الگوریتم پایان می پذیرد، در غیر این صورت، ذرات

---

<sup>۱</sup> Dissipative PSO (DPSO)

<sup>۲</sup> Divided rang PSO (DRPSO)

مجدداً جمع آوری می شوند و بار دیگر بر مبنای تابع هدف بعدی، دسته بندی می شوند. این الگوریتم برای خوشه بندی هوک<sup>۱</sup> و شبکه های تلفن همراه به کار گرفته می شود [۳۳].

### ۱۰.۴.۳. بهینه سازی توده ذرات قابل تنظیم و متغیر<sup>۲</sup>

به منظور ایجاد توازن بین اکتشاف و استخراج، در بهینه سازی توده ذرات و همچنین حفظ همگرایی ذرات، الگوریتم بهینه سازی توده ذرات قابل تنظیم و متغیر معرفی گردید که در آن، فاصله هر ذره از بهترین موقعیت برای تعیین سرعت هر ذره در هر تکرار استفاده می شود [۳۴].

### ۱۱.۴.۳. بهینه سازی توده ذرات با همسایگی متغیر<sup>۳</sup>

روش بهینه سازی توده ذرات با همسایگی متغیر، بهبودهایی نسبت به روش بهینه سازی توده ذرات عادی دارد. در این روش، به جای استفاده از بهترین پاسخ عمومی فعلی در الگوریتم، پارامتر دیگری با نام "بهترین در همسایگی" تعریف می شود. این پارامتر، بهترین ذره در میان همسایگی مشخص را انتخاب می کند. این روش روی این مسئله بحث می کند که انتخاب همسایگی برای ذره فعلی به عنوان یک هدف، آن را چند هدفه می سازد. علاوه بر این، انتخاب بهترین در میان آن ها، خود یک هدف دیگر است [۳۵].

---

<sup>۱</sup> HOC

<sup>۲</sup> Dynamic and Adjustable PSO (DAPSO)

<sup>۳</sup> Dynamic neighborhood PSO (DNPSO)

### ۱۲.۴.۳. تخمین توزیع بهینه سازی توده ذرات<sup>۱</sup>

این الگوریتم، ترکیبی از بهینه سازی توده ذرات و الگوریتم تخمین توزیع<sup>۲</sup> می باشد. همچنین، الگوریتم های تخمین توزیع از اطلاعات بدست آمده از مدل های آماری، که روی فضای پاسخ مناسب روی توزیع در طول فرآیند بهینه سازی تولید شده اند شکل می گیرد و تلاش می کند، مناطق بهتری را بیابد. این ویژگی در این الگوریتم، برای بهبود بازدهی بهینه سازی توده ذرات به کار گرفته می شود [۳۶].

### ۱۳.۴.۳. بهینه سازی توده ذرات گسترش یافته<sup>۳</sup>

در این الگوریتم، مزایای همزمان نسخه های بهترین پاسخ عمومی و بهترین پاسخ محلی به کار گرفته می شوند. در حقیقت، ترکیبی از این دو در رابطه به روزرسانی سرعت استفاده می شوند. تفاوت بین این الگوریتم ها با الگوریتم بهینه سازی توده ذرات کاملاً آگاهانه، در بهای محاسباتی کمتر می باشد [۳۷].

### ۱۴.۴.۳. بهینه سازی توده ذرات گاوسی<sup>۴</sup>

بهینه سازی توده ذرات تصویری از ذرات را ارائه می دهد که به دنبال نقطه بهینه هستند، این تصویر شبیه به رفتار زنبورهای است که پیرامون گل می گردند. یکی از اهداف، تصویری سازی گراف های توده است که در آن تمامی ذرات هر نسل (تکرار) حضور دارند. هدف دیگر، نمایش موفقیت آمیز

---

<sup>۱</sup> Estimation of Distribution PSO (EDPSO)

<sup>۲</sup> Estimation of Distribution Algorithm (EDA)

<sup>۳</sup> Extended PSO (EPSO)

<sup>۴</sup> Gaussian PSO (GPSO)

بهترین ها است، که در آن تحلیلی از نحوه عملکرد توده نشان می دهد. این اهداف فهم کمی از بهینه سازی توده ذرات به ما می دهد. این اهداف، هدف تصویری جدیدی مبتنی بر احتمال توزیع توده ارائه می دهد، بنابراین طبیعت تصادفی بهینه سازی توده ذرات به خوبی به تصویر کشیده می شود. این تصویرسازی فهم بهتری از چگونگی تنظیم الگوریتم و برطرف کردن ضعف ها می دهد. این الگوریتم بر اساس جابجایی توده بر اساس فاصله گاوسی از بهترین عمومی و محلی ارائه می دهد [۵۱].

### ۱۵.۴.۳. مدل بهینه سازی توده ذرات ژنتیک باینری<sup>۱</sup>

این الگوریتم برای افزایش شرایط دینامیک و توانایی اکتشاف در توده بهبود داده شد. در الگوریتم بهینه سازی توده ذرات باینری، پارامترهای مرگ و زایش به کار گرفته می شوند. به عبارت دیگر، بر مبنای اصول بهینه سازی توده ذرات باینری، موقعیت و سرعت به روزرسانی می شوند، سپس ذرات کودک به توده افزوده می شوند و برخی دیگر از بین می روند و از توده جدا می شوند. در حالت باینری برای در نظر گرفتن هر ذره به عنوان کروموزوم و زنجیره ای به ابعاد فضا هزینه ای ندارد [۳۹].

### ۱۶.۴.۳. بهینه سازی توده ذرات حریمانه<sup>۲</sup> یا بهینه سازی توده ذرات با ساختار دوگانه

#### کد شده باینری<sup>۳</sup>

این الگوریتم اولین بار برای حل مساله کوله پشتی مطرح گردید که بسیار هم موفق عمل کرد. این روش از مزایای روش تبدیل حریمانه بهره می گیرد. این الگوریتم یک الگوریتم تکاملی مرکب می

---

<sup>۱</sup> Genetic binary PSO (GBPSO)

<sup>۲</sup> Greedy PSO (GPSO)

<sup>۳</sup> Double-Structure coding Binary PSO (DS-BPSO)

باشد که بهینه سازی توده ذرات را با تبدیل حریصانه ترکیب می کند. علاوه بر این بهینه سازی توده ذرات باینری با ساختار دوگانه کد شده معرفی شده است [۴۰].

### ۱۷.۴.۳. بهینه سازی توده ذرات گروه دوستان<sup>۱</sup>

در این الگوریتم، ذرات تنها از دانش اجتماعی برای اکتشاف در فضای جستجو استفاده می کنند. اگر در بهینه محلی گیر کنند، یک بردار تصادفی سرعت به کار گرفته می شود. در این الگوریتم ها، بر خلاف بهینه سازی توده ذرات، تجمع نهایی استفاده می شود تا پارامترها به جواب برسند و این نوعی از خودتنظیمی برای پارامترها است [۴۱].

### ۱۸.۴.۳. بهینه سازی توده ذرات ابتکاری<sup>۲</sup>

در سال ۲۰۰۷ یک روش متفاوت از بهینه سازهای توده ذرات معرفی شد که بهینه سازی توده ذرات ابتکاری نامیده شد، که با نسخه اصلی الگوریتم متفاوت است و این تفاوت در نحوه انتخاب ذره بعدی برای به روز رسانی موقعیت و سرعت می باشد. هدف استفاده شده در این الگوریتم می تواند سرعت همگرایی به بهینه محلی توده را بهبود بخشد. برای جلوگیری از همگرایی ناخواسته، زمانی که ذرات نزدیک به بهترین نقطه عمومی باشند موقعیت ذرات مجدداً به طور تصادفی مقادری می شوند. ترکیب به روز رسانی ابتکاری و مقادری اولیه مجدد موقعیت به الگوریتم بهینه سازی توده ذرات ابتکاری امکان می دهد تا عملکرد بهتری نسبت به بهینه سازی توده ذرات پایه داشته باشد [۴۲].

---

<sup>۱</sup> Gregarious PSO (GPSO)

<sup>۲</sup> Heuristic PSO (HPSO)

### ۱۹.۴.۳. بهینه سازی توده ذرات با ساختار چند طبقه ای<sup>۱</sup>

این الگوریتم در سال ۲۰۰۴ معرفی گردید. در این روش، ذرات با استفاده از تعریف یک ساختار همسایگی در یک ساختار چند طبقه متغیر چیده می شوند. بسته به کیفیت آن ها نسبت به بهترین پاسخی که تا کنون یافت شده، ذرات به طبقات بالا یا پایین ساختار جابجا می شوند. علاوه بر این، الگوریتم دیگری با نام "بهینه سازی توده ذرات با ساختار چند طبقه ای دسته بندی شده"<sup>۲</sup> نیز در سال ۲۰۰۴ معرفی شد [۴۳].

### ۲۰.۴.۳. الگوریتم بهینه سازی توده ذرات مرکب با کاهش گرادیان<sup>۳</sup>

در این روش، دو عبارت سرعت، یکی به سمت بهینه عمومی و دیگری در خلاف جهت گرادیان، به کار گرفته می شوند. همچنین جابجایی در خلاف جهت گرادیان ممکن است به بهینه محلی منجر شود، سرعت جابجایی به سمت بهینه عمومی، می تواند از آن جلوگیری کند. محاسبات گرادیان، مجموع محاسبات را افزایش می دهد اما در عوض، محاسبات ذرات همسایه نادیده گرفته می شوند [۴۴].

### ۲۱.۴.۳. بهینه سازی توده ذرات مرکب با تبرید شبیه سازی شده<sup>۴</sup>

بهینه سازی توده ذرات در پیدا کردن بهینه عمومی بسیار موثر عمل می کند اما از طرفی ممکن است در بهینه محلی گیر کند. در عوض الگوریتم ژنتیک در یافتن بهینه محلی تواناست و می تواند از افتادن در آن جلوگیری کند، اما در یافتن بهینه عمومی ضعیف عمل می کند. بنابراین ترکیب این دو

---

<sup>۱</sup> Hierarchical PSO (HPSO)

<sup>۲</sup> Partitioned Hierarchical PSO (PH-PSO)

<sup>۳</sup> Hybrid gradient descent PSO (HGPSO)

<sup>۴</sup> Hybrid PSO with simulated annealing (SAPSO)



الگوریتم می تواند ضعف های آن ها را پوشش دهد. به منظور ترکیب این دو روش، پس از تولید ذرات تصادفی، تبرید شبیه سازی شده روی هر ذره اعمال می شود و سپس موقعیت و سرعت آن ها به روزرسانی می شود. این رویه ادامه می یابد تا زمانی که شرایط توقف برقرار شود [۴۵].

### ۲۲.۴.۳. بهینه سازی توده ذرات مکرر<sup>۱</sup>

در این روش یک رشته، با نام بهترین تکرار، به الگوریتم اصلی افزوده می شود تا کیفیت پاسخ و بازده محاسباتی را بهبود دهد. گسترش بهای تولید خطی، بهای قرارداد بازسازی، بهای قرارداد گنجایش تقاضا و قبض جریمه در انتخاب بهینه ظرفیت های قرارداد در نظر گرفته می شوند [۴۶].

### ۲۳.۴.۳. بهینه سازی توده ذرات کاهش نقشه<sup>۲</sup>

زمانی که مسائلی با مقادیر بزرگ داده را حل می کنیم، بهینه سازی توده ذرات ممکن است بازدهی لازم را نداشته باشد، چون تکامل هر تابع ممکن است زمان زیادی به طول انجامد. بنابراین، الگوریتم بهینه سازی توده ذرات با کاهش نقشه معرفی گردید، که در واقع اجرای موازی بهینه سازی توده ذرات برای توابع فشرده محاسباتی می باشد [۴۷].

### ۲۴.۴.۳. بهینه سازی توده ذرات باینری بهبود یافته<sup>۳</sup>

در این الگوریتم، که نسخه بهبود یافته ای از الگوریتم بهینه سازی توده ذرات باینری می باشد، تمامی ذرات به صورت بردارهای باینری و به صورت تصادفی تولید می شوند. سپس، حداقل مقدار موقعیت،

---

<sup>۱</sup> Iteration PSO (IPSO)

<sup>۲</sup> Map Reduce PSO (MRPSO)

<sup>۳</sup> Modified Binary PSO (MBPSO)

برای نگاشت فضای باینری به فضای جایگشت استفاده می شود. در این الگوریتم، روابط جدیدی برای به روز رسانی مقادیر موقعیت و سرعت به کار گرفته می شوند [۴۸].

### ۲۵.۴.۳. بهینه سازی توده ذرات ژنتیک بهبود یافته<sup>۱</sup>

این الگوریتم در حقیقت ترکیبی از دو الگوریتم بهینه سازی توده ذرات ژنتیک و الگوریتم های تکاملی تفاضلی می باشند. در این الگوریتم، تلاش شده تا بازدهی بهینه سازی توده ذرات ژنتیک بهبود یابد. به روزرسانی موقعیت بعدی برای هر ذره توسط هر دو الگوریتم صورت می گیرد و نتیجه بهتر برای جابجایی بعدی ذره در نظر گرفته می شود [۴۹].

### ۲۶.۴.۳. بهینه سازی توده ذرات عصبی<sup>۲</sup>

در این روش، یک شبکه عصبی مستقیم با بهینه سازی توده ذرات ترکیب می شود، ذرات عصبی در فضایی شبیه به فضای شبکه عصبی مستقیم معرفی می شوند. در این شبکه عصبی، گام آموزش جابجایی ذرات برای یافتن بهترین جایگاه در فضا می باشد [۵۰].

### ۲۷.۴.۳. الگوریتم بهینه سازی جدید مبتنی بر بهینه سازی توده ذرات<sup>۳</sup>

در این روش، عبارت جدید با نام "ممانتم" استفاده می شود، که به عنوان اثر مشخصه همگرایی الگوریتم بهینه سازی توده ذرات اصلی شناخته شده است. نشان داده شده است که ساختار الگوریتم

---

<sup>۱</sup> Modified Genetic PSO (MGPSO)

<sup>۲</sup> Neural PSO (NPSO)

<sup>۳</sup> New optimization algorithm based on PSO

جدید، مسائلی که توسط بهینه سازی توده ذرات همگرایی ضعیفی دارند را به خوبی حل می کند و همچنین می تواند فرآیند جستجوی بهینه ذرات را به درستی به انجام برساند [۵۱].

### ۲۸.۴.۳. بهینه سازی توده ذرات جدید<sup>۱</sup>

در این الگوریتم، برای محاسبه بهترین عمومی و بهترین شخصی به جای استفاده از بهترین از بدترین استفاده می شود. هرچند، این عبارات، با علامت منفی در روابط به روزرسانی سرعت استفاده می شوند. به عبارت دیگر، استفاده از این فرآیند به جای نزدیک شدن به بهترین نقاط سعی می کند از بدترین نقاط فاصله بگیرد [۵۲].

### ۲۹.۴.۳. بهینه سازی توده ذرات نیچینگ<sup>۲</sup>

این الگوریتم برای بهینه سازی های چندگانه مورد استفاده قرار می گیرد. در این الگوریتم، اهداف تجاری در ابتدا قرار می گیرند. در طول فرآیند اجرای بهینه سازی توده ذرات، ذرات به صورت فردی بررسی می شوند. ذره ای که در طول هر تکرار، مقدار تطبیق آن کمی تغییر می کند یا به کل تغییر نمی کند از توده اصلی جدا می شود و در یک زیرگروه قرار می گیرد. همین طور که الگوریتم به پیش می رود، اعضای توده اصلی کاهش می یابند و زیرگروه های جدیدی تولید می شوند. در حقیقت، این زیرگروه ها به صورت متغیر ساخته می شوند، تا بهینه های محلی و عمومی را به صورت همزمان و موازی بیابند [۵۳].

---

<sup>۱</sup> New PSO (NPSO)

<sup>۲</sup> Niching PSO

### ۳۰.۴.۳. بهینه سازی توده ذرات مرکب جدید<sup>۱</sup>

این الگوریتم، بازده بهینه سازی توده ذرات برای حل مسائل با ابعاد زیاد را بهبود بیشتری می دهد. این روش ترکیبی از بهینه سازی توده ذرات و طرح جستجوی هماهنگ<sup>۲</sup> می باشد. همچنین طرح جستجوی هماهنگ در جستجو به الگوریتم بهینه سازی توده ذرات کمک می کند و نتیجه این امر، بهبود قدرت اکتشاف بهینه سازی توده ذرات می باشد [۵۴].

### ۳۱.۴.۳. بهینه سازی توده ذرات متعامد<sup>۳</sup>

در این الگوریتم، به منظور به روزرسانی سرعت، سیستمی با نام روش جابجایی هوشمند<sup>۴</sup> به جای سیستم همرفتی استفاده می شود. در حقیقت، در روش جابجایی هوشمند، هدف تقسیم و پیروزی برای تعیین حرکت بعدی ذره استفاده می شود. نتایج مبین این امر هستند که این الگوریتم در مسائل با پارامترهای بزرگ نتایج بهتری نسبت به الگوریتم اصلی دارد [۵۵].

### ۳۲.۴.۳. بهینه سازی توده ذرات موازی<sup>۵</sup>

این الگوریتم در سال ۲۰۰۵ معرفی گردید. در این الگوریتم، نیازمندی های زمانی برای حل مسائل پیچیده و با ابعاد زیاد مهندسی را می توان به طور موثری با محاسبات موازی کاهش داد. با توجه به تقاضا در یک مساله محاسباتی شناسایی سیستم بیومکانیکی، یک پیاده سازی از بهینه ساز عمومی

---

<sup>۱</sup> Novel Hybrid PSO (NHPSO)

<sup>۲</sup> Harmony search scheme

<sup>۳</sup> Orthogonal PSO

<sup>۴</sup> Intelligent Move Method (IMM)

<sup>۵</sup> Parallel PSO (PPSO)

مبتنی بر جمعیت، به عنوان ابزاری برای افزایش خروجی معرفی گردید. توده ذرات با توجه به سادگی به پارامترهای الگوریتمی بسیار کمی برای تعیین رفتار همگرایی نیاز دارد [۵۶].

### ۳.۴.۳. بهینه سازی توده ذرات موازی غیرهمزمان<sup>۱</sup>

این الگوریتم از الگوریتم بهینه سازی توده ذرات موازی نشات گرفته است. در الگوریتم بهینه سازی توده ذرات موازی غیرهمزمان، تفاوت در همزمانی و غیرهمزمانی در روابط به روزرسانی موقعیت و سرعت وجود دارد. به روز رسانی ذرات و سرعت به طور پیوسته با توجه به اطلاعات در دسترس انجام می گیرد. این الگوریتم نمایی متغیر از بار متوازن با هدف کاهش عدم توازن در بار را تولید می کند [۵۷].

### ۳.۴.۳. بهینه سازی توده ذرات موازی همزمان<sup>۲</sup>

این الگوریتم به روز رسانی موقعیت و سرعت ذرات را در انتهای هر تکرار به صورت کاملاً همزمان انجام می دهد. این روش از یک بار ثابت و متوازن استفاده می کند که در آن وظایف هر پردازنده در زمان مجموع تعیین شده است [۵۸].

### ۳.۴.۳. بهینه سازی توده ذرات موازی مبتنی بر بردار<sup>۳</sup>

این الگوریتم در واقع ترکیبی از چند الگوریتم شامل بهینه سازی توده ذرات، بهینه سازی توده ذرات نیچینگ، و بهینه سازی توده ذرات مبتنی بر بردار می باشد. این الگوریتم تلاش دارد تا زیرگروه های

---

<sup>۱</sup> Parallel Asynchronous PSO (PAPSO)

<sup>۲</sup> Parallel synchronous PSO (PSPSO)

<sup>۳</sup> Parallel vector-based PSO (PVPSO)

موجود در نیچینگ را به طور همزمان بهینه سازی نماید. همچنین تلاش می کند تا کل توده را به صورت همزمان به سمت بهینه همگرا نماید. بنابراین این روش، بر روی مسائل بهینه سازی همزمان چندین تابع اعمال می شود [۵۹].

### ۳.۴.۳. بهینه سازی توده ذرات با گسترش مناطق<sup>۱</sup>

این الگوریتم برای جابجایی تعدادی روبات در یک محوطه طراحی گردید. این الگوریتم در واقع شامل بهبود هایی در بهینه سازی توده ذرات همرفتی می باشد. این بهبودها، همراه با اطلاعاتی از فضای گسترش یافته می باشند. برای دستیابی به این هدف، مجموعه ای از روش های ابتکاری به کار گرفته می شوند تا سرعت ذرات را به روزرسانی کنند. هرچند، برخی روش های ابتکاری به کار گرفته شده برای جلوگیری از افتادن در بهینه های محلی و همچنین جلوگیری از گیر کردن مسئله می باشند [۶۰].

### ۳.۴.۳. بهینه سازی توده ذرات با رفتار فاصله ای<sup>۲</sup>

در این الگوریتم، فضای پرواز ذرات، به مناطق مختلفی تقسیم می شود. در نتیجه، توده رفتار ثابتی نخواهد داشت و با توجه به منطقه پرواز رفتار متفاوتی خواهد داشت. به همین دلیل است که در مناطق مورد نظر ذرات پرواز سریعتری به سمت بهترین موقعیت خواهند داشت و در سایر مناطق با سرعت عادی حرکت می کنند [۴۲].

---

<sup>۱</sup> PSO with area extension (AEP SO)

<sup>۲</sup> PSO with behavior of distance (BDPSO)

### ۳۸.۴.۳. بهینه سازی توده ذرات با آشفستگی و صعود از تپه<sup>۱</sup>

در اکثر الگوریتم های وابسته به بهینه سازی توده ذرات، تلاش شده تا توازی بین اکتشاف و استخراج در الگوریتم وجود داشته باشد. این توازن موفقیت مهمی است. این الگوریتم، برای بهینه سازی توابع چند حالته، از آشفستگی و صعود از تپه استفاده می کند تا به ترتیب اکتشاف و استخراج را بهبود بخشد [۶۱].

### ۳۹.۴.۳. بهینه سازی توده ذرات با سرعت فرار<sup>۲</sup>

این الگوریتم برای جلوگیری از همگرایی سریع، و در نتیجه برای افزایش تغییرات در توده می باشد. در واقع، یک سرعت فرار به همه ی ذرات افزوده می شود تا همانگونه که بیان شد اهداف مورد نظر قابل دستیابی باشند [۶۲].

### ۴۰.۴.۳. بهینه سازی توده ذرات با گردآوری غیرفعال<sup>۳</sup>

گروه بندی دارای دو نوع می باشد: (۱) هم فزونی<sup>۴</sup>: که خود شامل دو نوع می باشد. نوع اول هم فزونی غیرفعال که در آن یک گروه غیر فعال با یک فرآیند فیزیکی تشکیل می شود، مانند گروه های پلانکتون ها که در آب شناور هستند و جریان آب آن ها را در کنار هم نگه می دارد. نوع دوم هم فزونی فعال می باشد که در آن هم فزونی توسط یک منبع جذب کننده انجام می گیرد. این منبع جذب کننده می تواند غذا یا آب باشد. (۲) گردآوری: این گروه بندی متفاوت با هم فزونی می باشد. در

---

<sup>۱</sup> PSO with craziness and hill climbing (CPSO)

<sup>۲</sup> PSO with Escape Velocity (EVPSO)

<sup>۳</sup> PSO with passive congregation (PSOPC)

<sup>۴</sup> Aggregation

اینجا منبع جذب کننده عوامل و مسائل فیزیکی نمی باشد که گروه را با آن مجبور کند. این نوع نیز به دو دسته تقسیم می شود: (۱) گردآوری غیرفعال که در آن جذب از یک ذره به بقیه است اما به صورت یک رفتار اجتماعی نمی باشد. (۲) گردآوری اجتماعی که در آن رفتار اجتماعی در بین ذرات وجود دارد و آن ها قویاً به هم مرتبط می باشند. از آن جایی که در برخی گروه ها ممکن است رفتارهای خودخواهانه در به اشتراک گذاری اطلاعات وجود داشته باشد. به عنوان مثال در توده ماهی ها، یک رفتار خودخواهانه می تواند یک گروه غیرفعال را شکل دهد. مدل یک گروه غیرفعال را می توان به یک بهینه سازی توده ذرات برای افزایش کارآیی اضافه کرد. این عبارت همانند انتخاب تصادفی ذرات از توده و ذره فعلی می باشد [۶۳].

### ۴۱.۴.۳. بهینه سازی توده ذرات با گسترش فضایی ذرات<sup>۱</sup>

در این الگوریتم، ذرات مشابه در یک زیرگروه جمع آوری می شوند که گونه<sup>۲</sup> نامیده می شود. برای بررسی معیار شباهت ذرات، از فاصله اقلیدسی استفاده می شود. در این روش، ذرات در گونه خود حرکت می کنند و در حقیقت بهینه سازی چند هدفه موازی انجام می شود. با اجرای بهینه سازی توده ذرات، مجموع همگرایی در هر گونه افزایش می یابد [۶۴].

### ۴۲.۴.۳. بهینه سازی توده ذرات با تعقیب و گریز<sup>۳</sup>

این الگوریتم از ایده ماهی کوچک و نهنگ قاتل استفاده می کند. در این دو گروه ماهی کوچک در حال فرار و نهنگ قاتل در حال تعقیب می باشد. در شبیه سازی این رفتار در بهینه سازی توده ذرات،

---

<sup>۱</sup> PSO with spatial particle extension (SEPSO)

<sup>۲</sup> Species

<sup>۳</sup> Pursuit-Escape PSO (PEPSO)



توده در واقع به دو زیرگروه تقسیم می شود که به ترتیب گروه فرار کننده و گروه تعقیب کننده نامیده می شوند. این ایده باعث متمرکزسازی به واسطه گروه تعقیب کننده و غیر متمرکزسازی به واسطه گروه فرارکننده می شود. این ایده باعث جلوگیری از گیر کردن در بهینه محلی می شود. علاوه براین یک توازن مناسب بین متمرکزسازی (جستجوی محلی) و غیرمتمرکزسازی (جستجوی عمومی) بر قرار می سازد [۶۵].

### ۴۳.۴.۳. بهینه سازی توده ذرات مبتنی بر چاه پتانسیل کوانتوم دلتا<sup>۱</sup>

در بهینه سازی توده ذرات، یک ذره با مسیرش مشخص می گردد که در واقع مقادیر موقعیت و سرعت می باشند. اما در کوانتوم براساس اصل عدم قطعیت مسیر معنی پیدا نمی کند چون موقعیت و سرعت را نمی توان به صورت همزمان تعریف کرد. بنابراین، تعاریف در کوانتوم متفاوت هستند، با این وجود اصول بهینه سازی توده ذرات حفظ می شوند [۶۶].

### ۴۴.۴.۳. بهینه سازی توده ذرات تحت تاثیر کوانتوم<sup>۲</sup>

این الگوریتم به همه ذرات اجازه می دهد تا در همه ی نسخه های بهینه سازی توده ذرات به جای تغییرات کلاسیک نیوتونی رفتار کوانتومی داشته باشند. بنابراین، به جای قدم زدن تصادفی نیوتونی، حالتی از حرکت کوانتومی بر فضای جستجو اعمال می شود. زمانی که این الگوریتم روی مجموعه توابع آزمایش گردید، در مقایسه با بهینه سازی توده ذرات کلاسیک بازدهی قابل توجهی نشان داد اما در حالت اندازه جمعیت زیاد. یکی از مشخصه های جذاب الگوریتم جدید کاهش پارامترهای کنترلی می باشد [۶۷].

---

<sup>۱</sup> Quantum Delta-Potential-Well-Based PSO (QDPSO)

<sup>۲</sup> Quantum inspired version of the PSO (QPSO)

### ۴۵.۴.۳. بهینه سازی توده ذرات با درون یابی چهارتایی<sup>۱</sup>

این الگوریتم از یک عملگر چندگانه بازتولید یا ضرب چهارتایی استفاده می کند. عملگر بازتولید، در حقیقت از الگوریتم های تکاملی نشات گرفته شده است. این الگوریتم، از ایده داشتن چندین شریک، استفاده می کند. در این الگوریتم، رهبر توده در هر تکرار معین می گردد. سپس، شریک های آن در میان ذرات دیگر انتخاب می شوند. سپس، یک عملگر ضرب به کار گرفته می شود. ذره جدید اگر نتیجه بهتری نسبت به بهترین ذره موجود در توده داشته باشد در توده پذیرفته می شود [۶۸].

### ۴۶.۴.۳. بهینه سازی توده ذرات با سرعت محصور<sup>۲</sup>

الگوریتم بهینه سازی توده ذرات برای بهینه سازی مسائل نامحدود به کار می رود. مکانیزم جستجوی آن از نوعی است که در آن فضای جستجو نامحدود است. هرچند گاهی مسائلی که در آن فضای جستجو محدوده قابل قبولی دارد نیز وجود دارند. برای حل چنین مسائلی الگوریتم معرفی شده به کار گرفته می شود. در این دستیابی، سرعت ذره محدود در نظر گرفته می شود [۶۹].

### ۴۷.۴.۳. بهینه سازی توده ذرات با خود سازمان دهی<sup>۳</sup>

در این الگوریتم، علاوه بر اطلاعات ذره و اطلاعات کل توده، یک بازخورد نیز در نظر گرفته می شود تا بازده ذره را بهبود بخشد. همچنین، ذره، اطلاعات بازخورد کل توده را استفاده می کند، و مقدار آن تعیین می کند و برای تکرارهای بعدی رفتار آن را بهبود می بخشد. هدف عمده، جلوگیری از همگرایی نامطلوب در کل الگوریتم می باشد [۷۰].

---

<sup>۱</sup> Quadric Interpolation PSO (QLPSO)

<sup>۲</sup> Restricted Velocity PSO (RVPSO)

<sup>۳</sup> Self-organization PSO (SOPSO)

### ۴۸.۴.۳. گونه در یک بهینه ساز توده ذرات<sup>۱</sup>

در این روش، ذرات مشابه در زیر گروه های مختلف دسته بندی می شوند. شرایط شباهت، به عنوان مثال، می تواند فاصله اقلیدسی باشد. هرکدام از این زیر گروه ها یک گونه نامیده می شود. در هر گونه، ذره ای که بهترین تطبیق را دارد، بذر یا دانه ی گونه خواهد و برای آن یک قطر تعریف می شود. در حقیقت، هر ذره در گونه خود قرار می گیرد و تکامل های موازی انجام می شود. بنابراین، این روش برای مسائل بهینه سازی چند هدفه اعمال می شود [۷۱].

### ۴۹.۴.۳. بهینه سازی توده ذرات آموزش دیده<sup>۲</sup>

این الگوریتم از مکانیزمی استفاده می کند که سعی در کاهش پیچیدگی و زمان همگرایی دارد. این الگوریتم به شبکه های ارتباطی اد-هوک<sup>۳</sup> اعمال می شود. در حقیقت، ذرات در این شبکه جابجا می شوند. به زبان ساده، جابجایی این ذرات پیچیدگی محاسباتی را افزایش می دهد و بنابراین، زمان همگرایی را کاهش می دهد. این ضعف ها با آموزش ذرات برطرف می شوند [۷۲].

### ۵۰.۴.۳. بهینه سازی توده ذرات مبتنی بر دو توده<sup>۴</sup>

هدف این الگوریتم فرار از گیر کردن در بهینه محلی و جلوگیری از همگرایی سریع می باشد. دو توده با پارامتر های مختلف در فضا جاری می شوند. به این ترتیب، ذرات هر دو توده، مسیرهای متفاوتی نسبت به هم خواهند داشت. یکی از آن ها توانایی در یافتن بهینه عمومی را بهبود می دهد و دیگری

---

<sup>۱</sup> Species in a Particle Swarm Optimizer (SPSO)

<sup>۲</sup> Trained PSO (TPSO)

<sup>۳</sup> Add - hoc

<sup>۴</sup> Two-Swarm based PSO (TSPSO)

با استفاده از طرح انتخاب تصادفی مبتنی بر انتخاب چرخش رولت<sup>۱</sup>، اکتشاف محلی را بهبود می بخشد [۷۳].

### ۵۱.۴.۳. بهینه سازی توده ذرات غیر محدود<sup>۲</sup>

بهینه سازی توده ذرات با توجه به پارامترهای سرعت یا موقعیت آن به دسته های محدود و غیر محدود تقسیم می شود. الگوریتم عموماً مانند حالت کلاسیک، محدود می باشد. در این حالت دو رابطه به روزرسانی سرعت و موقعیت، در هر دو حالت یکسان می باشند. هرچند در حالت کلاسیک (محدود)، شرایط مرزی بالا و پایین برای موقعیت و سرعت وجود دارند. اما در حالت غیر محدود شرایط مرزی وجود ندارد [۷۴].

### ۵۲.۴.۳. بهینه سازی توده ذرات متحد شده<sup>۳</sup>

این الگوریتم به منظور استفاده همزمان از مزایای بهترین پاسخ عمومی و بهترین پاسخ محلی، در بهینه سازی توده ذرات به کار گرفته شده است. در حقیقت، روابط به روزرسانی سرعت به دو گروه تقسیم شده اند که هر گروه سرعت را بر اساس اطلاعات داده شده از بهترین عمومی و بهترین محلی، محاسبه می کند [۷۵].

---

<sup>۱</sup> Roulette-wheel-selection

<sup>۲</sup> Unconstrained PSO (UPSO)

<sup>۳</sup> Unified PSO (UPSO)

### ۵۳.۴.۳. بهینه سازی توده ذرات با همسایگی متغیر<sup>۱</sup>

این الگوریتم یک الگوریتم فرا ابتکاری می باشد و ترکیبی از بهینه سازی توده ذرات و جستجوی همسایگی متغیر می باشد. این الگوریتم روی مساله برنامه ریزی شغل-فروش چند هدفه اعمال شده است. در این الگوریتم از جستجوی همسایگی متغیر برای بهبود بهینه ساز محلی استفاده شده است. این طرح جستجو از گیر افتادن در بهینه محلی جلوگیری می کند، و این کار را با تکرار از نقطه شروع به بهینه محلی که بهتر از بهینه محلی فعلی است ادامه می دهد [۷۶].

### ۵۴.۴.۳. بهینه سازی توده ذرات با سرعت محدود<sup>۲</sup>

اگر سرعت حرکت ذرات در محدوده های مختلف محدود باشد، پاسخ های بهینه متفاوتی ممکن است به دست آید. بنابراین، با در نظر گرفتن حدود بالا و پایین برای سرعت و موقعیت، الگوریتم بهینه سازی توده ذرات با سرعت محدود به دست می آید. استراتژی این دستیابی، به نحوی است که فقط ذراتی که حدود را رعایت می کنند حفظ می شوند و مابقی حذف می شوند [۷۷].

### ۵۵.۴.۳. بهینه سازی توده ذرات عمودی<sup>۳</sup>

از آنجایی که در برخی تکرارهای بهینه سازی توده ذرات، نقطه بهینه عمومی در الگوریتم بهبود نمی یابد و ذره نزدیک به همان بهینه باقی می ماند و از آنجا که حرکت بعدی ذره، به شدت بستگی به حرکت قبلی دارد، حرکت در آن مسیر به سختی صورت می گیرد و به سمت بهینه محلی حرکت می

---

<sup>۱</sup> Variable Neighborhood PSO (VNPSO)

<sup>۲</sup> Velocity Limited PSO (VLPSO)

<sup>۳</sup> Vertical PSO (VPSO)

کند. بنابراین، الگوریتم پیشنهادی به نحوی بهبود یافته است که ذرات در دو جهت عمومی و عمودی حرکت می کنند و مشکل موجود تا حدی برطرف شود [۷۸].

۴. جستجوی ممنوع

جستجوی ممنوع یک الگوریتم فرا ابتکاری است که مسیر جستجوی آگاهانه ی منطقه ای را، به راه حل هایی در ورای فضای بهینه سازی منطقه ای، هدایت می کند. یکی از اجزای اصلی جستجوی ممنوع، استفاده از حافظه فعال می باشد، که باعث رفتار منعطف تر این الگوریتم می شود. استراتژی های مبتنی بر حافظه از مشخصات روش جستجوی ممنوع می باشند.

#### ۱.۴. معرفی

حالت ابتدایی جستجوی ممنوع از ایده ای از فرد گلوور<sup>۱</sup> (۱۹۸۶-۱۹۷۷) گرفته شده است. جستجوی ممنوع بر مبنای محوطه و فضای حل مساله شکل می گیرد و به منظور کیفیت گذاری بر هوش الگوریتم، باید حافظه فعال و جستجوی پاسخ را نیز در نظر داشت. ویژگی حافظه فعال در جستجوی ممنوع، امکان پیاده سازی رویه هایی، که قابلیت جستجوی اقتصادی و موثر فضای حل مساله را دارند را فراهم می کند. از آنجایی که انتخاب ها با توجه به اطلاعات جمع آوری شده در طول جستجو هدایت می شوند، جستجوی ممنوع در تضاد با طراحی های بدون حافظه ای است که به طور زیادی وابسته به پروسه های تصادفی هستند و حالتی از نمونه برداری را پیاده سازی می کنند. توجه عمده، بر پاسخ دهی موثر در جستجوی ممنوع می باشد، که در پیاده سازی های محتمل و غیر قابل کنترل از نظریه هایی حاصل می شود که در آن ها یک انتخاب بد، اغلب منجر به اطلاعات بیشتری نسبت به یک انتخاب تصادفی خوب می شود و بنابراین مبنایی برای استراتژی هایی به وجود می آورد که به طور موثر از مزایای گذشته جستجو بهبود یافته اند. جستجوی ممنوع را می توان به طور مستقیم به هر نوع مساله بهینه سازی اعمال کرد [۷۹].

---

<sup>۱</sup> Glover Fred



## ۲.۴. روش شناسی جستجوی ممنوع

در شناخته شده ترین حالت، جستجوی ممنوع را می توان مانند حالت ابتدایی جستجوی محلی عادی یا جستجوی همسایگی در نظر گرفت، انجام مکرر از یک نقطه به نقطه دیگر تا زمانی که شرایط مساله برقرار شود. هر پاسخ  $x$  دارای همسایگی می باشد که  $N(x) \subset X$ ، و هر پاسخ  $x' \in N(x)$  با یک نداشت از  $x$  حاصل شده است.

روش جستجوی ممنوع را می توان در مقابل روش گرادیان ساده قرار داد که هدف آن حداقل سازی تابع  $f(x)$  می باشد. این چنین روشی، فقط اجازه می دهد تا به پاسخ هایی در همسایگی حرکت کنیم و مقادیر تابع را بهبود بخشیم و زمانی که پاسخ مناسبی یافت نشود، الگوریتم پایان می یابد.  $x$  نهایی حاصل از روش گرادیان، بهینه ی منطقه ای نامیده می شود، چون در همسایگی خود، حداقل شرایط یکسان و یا بهتری نسبت به سایر پاسخ ها دارد. ضعف واضح یک روش گرادیان، در این است که بهینه منطقه ای، در اکثر حالات پاسخ بهینه ی کلی نخواهد بود و عموماً تابع  $f(x)$  روی کل بازه  $x \in X$  حداقل نخواهد بود. جستجوی ممنوع حرکت هایی را مجاز می داند که حالت فعلی تابع را خراب کنند و حرکت هایی را انتخاب می کند که از همسایگی تعمیم یافته  $N^*(x)$  باشند. ساختار حافظه های کوتاه و بلند همسایگی  $N^*(x)$  را شکل می دهند. به بیان دیگر، همسایگی تعمیم یافته، نتیجه نگهداری یک تاریخچه، از حالت های امتحان شده در طول جستجو می باشد. در استراتژی های جستجوی ممنوع مبتنی بر فرضیات کوتاه، خصوصیات  $N^*(x)$  زیر مجموعه ای از  $N(x)$  می باشد، و دسته بندی ممنوع برای شناسایی المان هایی از  $N(x)$  که از  $N^*(x)$  استخراج شده اند به کار می رود [۸۰].

## ۳.۴. ویژگی های الگوریتم جستجوی ممنوع

### ۱.۳.۴. لیست ممنوع

لیست ممنوع، لیستی از تعداد کمی از جابجایی‌ها (پاسخ‌ها) است. حافظه جابجایی‌ها می‌تواند مبتنی بر تاخر یا فرکانس باشد. حافظه کوتاه، مبتنی بر تاخر، با قرار دادن  $T$  جابجایی آخر به عنوان ممنوع، از تکرار حول یک همسایگی محلی در فضای پاسخ جلوگیری می‌کند. حرکات اخیر، در یک مکانیزم ذخیره می‌شوند که به لیست جابجایی‌های ممنوع اشاره دارد. تعداد جابجایی‌های درون لیست با توجه به ابعاد لیست ممنوع تعیین می‌شود، و با  $T$  نشان داده می‌شود. لیست، بر اساس قانون اولین ورودی – اولین خروجی عمل می‌کند. اطلاعات موخر دیگری که در لیست ممنوع ذخیره می‌شود ویژگی‌های پاسخ است. مقدار بزرگتر برای  $T$ ، جابجایی‌ها و پاسخ‌ها را برای زمان بیشتری در لیست ممنوع نگه می‌دارد.

حافظه بلند مبتنی بر فرکانس این امکان را فراهم می‌کند که جستجوها در بیشترین همسایگی‌ها هدایت شود. حافظه مبتنی بر فرکانس اطلاعات مازادی را در این زمینه که برای یک جابجایی ممنوع چند بار تلاش شده است را نیز در بر دارد.

### ۲.۳.۴. لیست نمایندگی

جستجوی ممنوع از لیست نمایندگی استفاده می‌کند تا لیستی از حرکات را برای بررسی مناسب بودن تهیه کند. یکی از جابجایی‌ها (پاسخ‌ها) لیست نمایندگی انتخاب می‌شود تا در پروسه جستجو بررسی شود. لیست نمایندگی نقش مهمی در بازدهی جستجوی ممنوع ایفا می‌کند.

### ۳.۳.۴. متمرکزسازی و غیر متمرکزسازی

دو استراتژی مبتنی بر حافظه اصلی بنیادی از جستجوی ممنوع را شکل می دهند. جندرو<sup>۱</sup> (۲۰۰۳) غیرمتمرکزسازی را در طراحی یک جستجوی ممنوع تنها عامل مهم و عمده به دست آورد. با استفاده از استراتژی متمرکزسازی نواحی پیرامون پاسخ های مطلوب بیشتر مورد جستجو قرار می گیرند و به طور عام، با راه اندازی مجدد، یک جستجو از پاسخ به دست آمده قبلی صورت می گیرد، تا به پاسخ مطلوب برسد. راه اندازی مجدد از لیست نمایندگی، که مناطق مورد نظر را بیان می کند به دست می آید. غیرمتمرکزسازی از طرفی دیگر، پروسه ی جستجو را به سمت بررسی مناطق مشاهده نشده پیش می برد و پاسخ هایی تولید می کند که با پاسخ های به دست آمده از روش های دیگر متفاوت می باشد.

### ۴.۳.۴. تابع خطای هدف

تابع هدف یک پاسخ مانند  $s$ ، با  $f_1(s)$  نمایش داده می شود و با رابطه زیر محاسبه می شود:

$$f_1(s) = \sum_{Tours} \sum_{Routs} t + \sum_{customers} a_i y_i \quad (۱-۴)$$

در محاسبه تابع هدف،  $\alpha_i$  بیانگر آخرین جریمه برای مشتری  $i$  می باشد، در حالی که  $\alpha_i = \max\{0, a_i - l_i\}$  می باشد. زمان نهایی رسیدن به مشتری  $i$  با  $\alpha_i$  نمایش داده می شود، در حالی که  $l_i$  آخرین زمان مجاز رسیدن به مشتری  $i$  می باشد. طراحی الگوریتم تضمین می کند که شرط  $\alpha_i \leq l_i + L_i^{max}$  برقرار باشد، که در آن بیشترین مقدار قابل قبول برای مشتری  $i$  می باشد.

---

<sup>۱</sup> Jendro

#### ۵.۳.۴. شرایط توقف

جستجو زمانی که بیشترین تعداد تکرار الگوریتم اصلی جستجوی ممنوع رسیده باشد متوقف می شود. یک شرط توقف، می تواند از قبل تعیین شود تا تعداد تلاش ها برای دستیابی به پاسخ در لیست پاسخ های ممنوع، به عنوان پاسخ جدید به دست آید. این موضوع، بیانگر این مطلب است که جستجو زمانی که به یک بهینه محلی رسید متوقف می شود.

#### ۴.۴. الگوریتم ممنوع

برای هر پاسخ اولیه، یک نقطه تصادفی  $I_i^*$  تعریف می شود. نقاط باقی مانده برای ایجاد یک پاسخ اولیه با استفاده از پاسخ های اولیه الگوریتم استفاده می شوند. سپس، نقطه شناسایی شده  $I_i^*$  به موقعیت اولیه بازمی گردد. پاسخ مجموعه ای از پاسخ های اولیه  $S = \{s_1, s_2, s_3, \dots, s_I\}$  می باشد. روال تمام پاسخ های اولیه، در یک حافظه فعال ذخیره می شوند و به همراه آن، مقدار تابع هدفی که پاسخ های اولیه روال از آن ها ناشی می شود نیز ذخیره می شوند. تمام روال هایی که فقط شامل یک نقطه هستند از حافظه فعال حذف می شوند.

شبه کد الگوریتم جستجوی ممنوع:

- مقداردهی اولیه شرایط توقف
- مقداردهی اولیه پارامترهای جستجوی ممنوع
- تولید همسایگی اولیه
- تولید تابع هدف برای همسایگی
- انتخاب بهترین مقدار همسایگی

- به روز رسانی لیست ممنوع

- پایان

- شروع بهینه سازی

- برای تعداد تکرارهای جستجوی ممنوع

- همسایگی نقطه بهینه فعلی را تولید می کنیم

- عناصر همسایگی را در لیست ممنوع مورد بررسی قرار می دهیم

- تابع هدف و مقادیر خطا را برای عناصر غیر ممنوع تولید می کنیم

- بهترین مقدار خطا را به عنوان مرکز همسایگی در تکرار بعدی انتخاب می کنیم

- لیست ممنوع را به روز رسانی می کنیم

- پایان

- پایان

دسته گسترده ای از مسائل دنیای واقعی هستند که می توان آن ها را توسط الگوریتم جستجوی ممنوع

حل کرد، تعدادی از آن ها در زیر بیان شده اند اما فقط به همین موارد محدود نمی شوند:

- مساله فروشنده دوره گرد
- برنامه ریزی شیفت پرستاری
- شبکه های عصبی
- مساله کوله پشتی
- شبکه های مخابراتی
- طراحی شبکه
- بهینه سازی عمومی
- برنامه ریزی
- تخصیص چهارتایی
- تخصیص فرکانسی
- اجاره ی اتوموبیل
- برنامه ریزی منابع پروژه
- مساله مسیر یابی وسایل نقلیه
- رنگ آمیزی گراف

## ۵.۴. نمونه هایی از ترکیب بهینه سازی توده ذرات و جستجوی ممنوع

### ۱.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای مسئله برنامه

#### ریزی کار و تامین اجتماعی انعطاف پذیر

مسئله برنامه ریزی کار و تامین اجتماعی<sup>۱</sup> در بسیاری زمینه ها مانند مدیریت تولید و بهینه سازی های مرکب بسیار مهم می باشد. مسئله برنامه ریزی کار و تامین اجتماعی، دارای دو مشکل عمده می باشد با نام های مسئله گمردن ماشین و مسئله ترتیب عمل. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای حل این مسئله، به منظور کاهش بیشینه زمان تکمیل معرفی شده است. در این الگوریتم مرکب، بهینه سازی توده ذرات به منظور تولید یک توده با کیفیت بالا، از کاندیداهای پاسخ به کار می رود، در حالی که جستجوی ممنوع، برای به دست آوردن پاسخ بهینه در نزدیکی پاسخ های مناسب به کار می رود. در بهینه سازی توده ذرات ساده، هر ذره از بهترین پاسخ محلی و بهترین پاسخ عمومی آموزش می بیند. بهینه سازی توده ذرات ساده برای مسائل پیوسته به کار می رود، اما مسئله برنامه ریزی کار و تامین اجتماعی، یک مسئله گسسته مرکب می باشد، بنابراین تغییراتی در بهینه سازی توده ذرات ساده باید انجام گیرد تا بتوان برای حل این مسئله از آن استفاده کرد. در بخش جستجوی ممنوع، بهترین های عمومی مورد بررسی قرار می گیرند تا از افتادن در حلقه و یا گیر کردن در بهینه محلی جلوگیری شود [۸۱].

---

<sup>۱</sup> Job - Shop

#### ۲.۵.۴. بهینه سازی توده ذرات مبتنی بر ایده جستجوی ممنوع

در بهینه سازی توده ذرات، زمانی که یک ذره بهترین پاسخ محلی را می یابد، تمامی ذرات حول یک پاسخ جمع می شوند و نمی توانند از آن فرار کنند. از طرفی، جستجوی ممنوع می تواند با دور شدن از بهترین پاسخ فعلی از پاسخ های بهینه محلی فرار کند. روش پیشنهادی ترکیب بهینه سازی توده ذرات و جستجوی ممنوع، روشی است که مزایای هر دو الگوریتم بهینه سازی توده ذرات و جستجوی ممنوع را با هم دارد. در این روش توده کلی به دو قسمت تقسیم می شود که، توده ۱ و توده ۲ نامیده می شوند. و این توده ها نقش های مهمی در متمرکز سازی و غیر متمرکز سازی بازی می کنند. توده ۱، وظیفه متمرکز سازی جستجو ها حول بهترین پاسخ فعلی را با استفاده از بهینه سازی توده ذرات بر عهده دارد، و توده ۲، وظیفه غیر متمرکز سازی را به منظور جلوگیری از افتادن در پاسخ بهینه محلی برای یافتن بهینه عمومی بر عهده دارد.

در فرآیند جستجوی الگوریتم بهینه سازی توده ذرات، برای زمانی که بهترین پاسخ های حال حاضر، در مدت تناوب معین در ابعاد مشخص، به روز رسانی نشده اند، ذرات حول پاسخ در ابعاد مشخص متمرکز می شوند. در این زمان، نتایج به بهینه عمومی رسیده اند یا در بهینه محلی گیر کرده اند. سپس ذرات توده ۱، موقعیت خود را حول بهترین مقدار عمومی تثبیت می کنند در حالی که هنوز اطلاعات به روزرسانی نشده اند، و توده ۲، جستجو را برای دور شدن از بهترین های محلی و عمومی انجام می دهد. با این حرکت ذرات، امکان اکتشاف پاسخ های بهتر متفاوت با پاسخ فعلی را دارا می باشند. بدین ترتیب، الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع، عمل متمرکز سازی و غیر متمرکز سازی را با استفاده از این دو توده انجام می دهد. سپس اگر دوره تناوب ممنوع سپری شده باشد، حالت ممنوع از بین می رود و به الگوریتم بهینه سازی توده ذرات باز می گردد و فرآیند جستجو تکرار می شود [۸۲].

### ۳.۵.۴. الگوریتم بهینه سازی مرکب بهینه سازی توده ذرات و جستجوی ممنوع

بهینه سازی توده ذرات ساده، یک روش جستجوی عمومی است در حالی که جستجوی ممنوع یک روش جستجوی محلی می باشد. در اینجا، ترکیبی از بهینه سازی توده ذرات و جستجوی ممنوع به منظور ارائه الگوریتم ترکیبی بهینه سازی توده ذرات و جستجوی ممنوع ارائه می شود که توانایی جستجوی محلی و عمومی را به طور همزمان در هر تکرار داشته باشد و احتمال یافتن بهینه های نهایی را بهبود بخشد و به طور موثری از گیر کردن در بهینه های محلی جلوگیری کند.

روش پیشنهادی الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع، توانایی اکتشاف بهینه سازی توده ذرات و توانایی اکتشاف جستجوی ممنوع را دارد و ضعف های هر دو را به حداقل می رساند. در این روش در هر تکرار نیمی از داده ها توسط بهینه سازی توده ذرات تحلیل می شوند و نیمی دیگر توسط جستجوی ممنوع مورد بررسی قرار می گیرند. در انتهای هر تکرار نتایج با هم ترکیب می شوند، متغیرها به روز رسانی می شوند و تا زمانی که شرایط توقف برقرار نباشد تکرارها ادامه می یابد [۸۳].

### ۴.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای انتخاب ژن،

#### برای کلاسه بندی تومورها با استفاده از داده های تعریف ژن

داده های تعریف ژن توسط هزاران یا ده ها هزار ژن، شناسایی می شوند. این امر می تواند باعث تطبیق مطلق یا شکست در تحلیل داده شود. انتخاب ژن، یکی از اجزای اصلی در سیستم های کلاسه بندی تومور مبتنی بر تعریف ژن می باشد. در اینجا روش مرکب بهینه سازی توده ذرات و جستجوی ممنوع، برای انتخاب ژن در کلاسه بندی تومور معرفی شده است. مزیت جستجوی ممنوع، به عنوان فرآیند بهبود محلی، کمک می کند تا الگوریتم در بهینه محلی گیر نکند و بازده را بهبود می دهد.



بهینه سازی توده ذرات، مزیت هایی، شامل مشخصه همگرایی خوب را دارا می باشد و در بهینه سازی موثر عمل می کند و پس از چند تکرار، همگرایی کاهش می یابد و الگوریتم بهینه سازی توده ذرات، می تواند منجر به یک همگرایی نامطلوب، به یک بهینه محلی شود. جستجوی ممنوع، یک روش بهینه سازی قدرتمند می باشد، که به صورت تئوری به یک پاسخ بهینه عمومی، همگرا می شود، اما زمان بیشتری برای رسیدن به نزدیکی بهینه عمومی نیاز دارد. روش ترکیبی، جستجوی ممنوع را روی بهینه سازی توده ذرات پیاده سازی می کند تا فرآیند جستجوی محلی را بهبود بخشد تا همگرایی را بهبود بخشد و از گیر کردن در بهینه محلی جلوگیری کند [۸۴].

#### ۵.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای برنامه ریزی

##### کارخانجات فولاد مبتنی بر مدیریت معماری تولید - ذخیره و تولید - تقاضا

در این طرح، رویه ای برای برنامه ریزی در جزئیات و ایجاد یک مدل برنامه نویسی غیر خطی برای حل مساله ارائه می شود. این مدل، به طور همزمان، برنامه ریزی تولید و تطابق لیست موجودی را دارا می باشد، و توابع چندگانه را پوشش می دهد، مانند بهای کل تاخیر، خطای تاخیر در پنجره زمان ارسال، تولید، تطابق لیست، خطای لغو سفارش.

به منظور حل این مسئله غیر خطی، روش مرکب الگوریتم بهینه سازی توده ذرات و جستجوی ممنوع به کار گرفته شده است، که در آن قوانین ابتکاری به منظور از بین بردن ضعف در پاسخ ها به کار گرفته شده است و سپس تنظیمات مربوط به پارامتر های الگوریتم بهینه سازی توده ذرات و الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع تحلیل می شود [۸۵].

#### ۶.۵.۴. الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع برای بهینه سازی

##### مسائل غیر خطی

این ترکیب به منظور جلوگیری از همگرایی نامطلوب الگوریتم بهینه سازی توده ذرات، به سمت بهینه های محلی نامطلوب می باشد. جستجوی ممنوع، یک روش فرا ابتکاری موثر می باشد که با استفاده از ساختار های حافظه فعال جستجوی را بهبود می بخشد. در این ترکیب از بهینه سازی توده ذرات و جستجوی ممنوع از دو لیست جستجوی ممنوع در بهینه سازی توده ذرات استفاده می کنیم:

اولین لیست جستجوی ممنوع، که لیست غیر عملی نامیده می شود، یک لیست مکمل برای کنترل پاسخ های غیرممکن می باشد. پاسخ های غیرممکن در این لیست، با ابعاد ثابت ذخیره می شوند. زمانی که یک موقعیت جدید برای یک ذره تعیین شد بررسی می شود که آیا این مقادیر در لیست غیر عملی وجود دارد یا خیر، اگر در لیست وجود داشته باشد، نادیده انگاشته می شود و در غیر این صورت، موقعیت ذره به روزرسانی می شود. هدف از استفاده لیست غیر عملی، جلوگیری از تکرار پاسخ های بررسی شده می باشد. دومین لیست ممنوع برای فرآیند غیرمتمرکزسازی استفاده می شود. این الگوریتم بهینه سازی را روی بهترین های محلی انجام می دهد [۸۶].

۵. الگوریتم پیشنهادی ترکیب بهینه سازی توده ذرات و جستجوی ممنوع  
برای آموزش شبکه عصبی

## ۱.۵. تعریف مساله

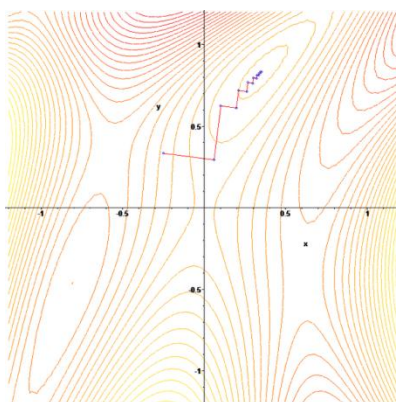
استفاده از یک شبکه عصبی در حل یک مساله، مستلزم استفاده از ساختاری شبکه ای می باشد، برای این منظور برای طراحی ارتباط میان نرون ها در لایه های مختلف شبکه عصبی، معماری و طرحی مانند یک شبکه در نظر گرفته می شود که می تواند حالت ها و طراحی های مختلفی داشته باشد. آموزش شبکه عصبی، مترادف با تقریب تابع می باشد، به این مضمون که، در آموزش شبکه عصبی از روی نمونه ها، شبکه با توجه به دانش حاصل از مجموعه نمونه ها، توانایی یافتن تابع حاکم بین ورودی و خروجی را دارا می باشد. در این حالت، پس از انجام گام آموزش، به ازای هر ورودی، با توجه به شرایط به دست آمده در گام آموزش، مقدار خروجی متناظر با آن قابل پیش بینی خواهد بود.

یکی از انواع شبکه های عصبی که در زمینه تقریب توابع کاربرد بسیاری دارد، شبکه عصبی توابع پایه شعاعی می باشد، این توابع، در ابتدا برای حل مسائل درونیایی چند متغیره حقیقی معرفی شده اند و برای اولین بار توسط برومهد و لو اولین در طراحی شبکه های عصبی مورد استفاده قرار گرفتند. آموزش لایه مخفی شبکه های عصبی توابع پایه شعاعی را می توان توسط الگوریتم های آموزشی ناظر انجام داد. برای این منظور، یک الگوریتم مبتنی بر گرادیان نزولی می توان در نظر گرفت و به کمک آن، وزن های هر نرون در شبکه را، تعیین کرد.

گرادیان نزولی، این قابلیت را دارا می باشد که در فضاهایی با هر ابعادی پیاده سازی شود، حتی در فضاهایی با ابعاد نامحدود. در حالتی که فضایی با ابعاد نامحدود داشته باشیم، فضای جستجو، همان فضای تابع می باشد. گرادیان نزولی می تواند، هر تعداد تکرار دلخواه برای محاسبه کمینه محلی انجام دهد تا به دقت مطلوب دست یابد. گرادیان نزولی را می توان با یک جستجوی خطی ترکیب نمود، تا بتوان میزان

مطلوب اندازه گام را در هر تکرار تعیین نمود. اجرای جستجوی خطی می تواند زمان بر باشد و انتخاب یک مقدار ثابت و کوچک برای اندازه گام نیز، می تواند منجر به همگرایی بسیار ضعیف شود.

با وجود ویژگی های گرادیان نزولی، این الگوریتم دارای ضعف هایی نیز می باشد، که از آن جمله می توان به این مساله اشاره کرد که در برخی کاربردها، گرادیان نزولی در نزدیکی نقطه کمینه کند می شود و عملاً سرعت همگرایی آن کاهش می یابد. در برخی موارد ممکن است، در نزدیکی نقطه کمینه با توجه به عمودی بودن مسیر حرکت گرادیان نسبت به خطوط کانتور، مانند شکل (۵-۱)، حالت زیگزاگی پیدا کند.



شکل ۵-۱: زیگزاگی شدن مسیر حرکت گرادیان در نزدیکی نقطه کمینه

به همین دلایل به دنبال جایگزینی مناسب برای الگوریتم گرادیان نزولی هستیم. در این زمینه الگوریتم های ابتکاری و فرا ابتکاری که در زمینه بهینه سازی معرفی شده اند گزینه های مناسبی می باشند.

چند سالی بیشتر از عمر الگوریتم های ابتکاری و فرا ابتکاری مانند بهینه سازی توده ذرات و جستجوی ممنوع نمی گذرد. اما با این وجود این الگوریتم ها بسیار متداول گشته اند و کاربردهای بسیاری در مسائل بهینه سازی دارند، از این جمله می توان شناسایی و تخمین سیستم ها و تقریب توابع را نام برد.

بهینه سازی توده ذرات و جستجوی ممنوع، از روش های محاسباتی بهره می گیرند که به طرز چشمگیری، میزان محاسبات را به ویژه برای مسائلی با ابعاد بالا کاهش می دهند. یکی از ویژگی های مهم این الگوریتم ها، توانایی آن ها در فرار از بهینه های محلی و حرکت به سوی کمینه مطلوب می باشد.

بهینه سازی توده ذرات اولین بار توسط کندی و ابرهارت، به عنوان روش بهینه سازی عمومی معرفی گردید. این الگوریتم توسط مجموعه ای از ذرات، که هر کدام نماینده یک پاسخ هستند، در فضای مسئله به جستجو می پردازد. ذرات با تاثیر پذیری از اینرسی، بهترین پاسخ فعلی و بهترین پاسخ یافت شده، توسط کل توده موقعیت خود را به روز می کنند.

جستجوی ممنوع توسط گلاور معرفی گردید و یک الگوریتم بهینه سازی مبتنی بر تکرار است که، از بهینه های محلی دوری می کند. جستجوی ممنوع، همسایگی هایی تولید می کند و با به روز رسانی همسایگی و مقایسه آن با لیست نتایج مشاهده شده قبلی، به سمت هدف حرکت می کند. این لیست، لیست ممنوع نامیده می شود و از حلقه و تکرار پاسخ جلوگیری می کند.

نمونه هایی را که تاکنون برای ترکیب کردن الگوریتم بهینه سازی توده ذرات و جستجوی ممنوع ارائه شدند را بیان کردیم، در این کار روش متفاوتی از ترکیب بهینه سازی توده ذرات و جستجوی ممنوع را ارائه می دهیم و همچنین به جای استفاده از الگوریتم بهینه سازی توده ذرات ساده، از یکی از نسخه های بهبود یافته آن استفاده می کنیم تا به کمک آن شبکه عصبی پایه توابع شعاعی را آموزش دهیم.

## ۲.۵. نسخه بهینه شده الگوریتم بهینه سازی توده ذرات

در بهینه سازی توده ذرات، هر ذره در فضای جستجو به سمت مقدار هدف حرکت می کند. این حرکت به سمت هدف، با استفاده از اطلاعات به دست آمده از سایر ذرات، در طول حرکت های آن ها در هر تکرار،

به دست می آید. این اطلاعات، در واقع اطلاعات اجتماعی هستند، که شامل اکتشافات و تجربیات تمامی ذرات توده می باشند. تجربیات شخصی ذرات یا در واقع همان بهترین شخصی و اطلاعات اجتماعی، همان بهترین عمومی ذرات می باشد که در طول تکرارها تاکنون به دست آمده و پاسخی است که بیشترین تطبیق را با مقدار هدف دارد.

از اطلاعات بهترین های شخصی و بهترین عمومی برای تولید سرعت حرکت ذرات به سمت هدف استفاده می شود. در مسائل کران دار، با این محدودیت مواجه هستیم که ذرات، ممکن است از محدوده مورد نظر ما خارج شوند به همین دلیل یکی از کارهایی که به این منظور می توان انجام داد، کنترل کردن سرعت، توسط محدود کردن آن به یک سرعت بیشینه می باشد، تا سرعت زیاد، باعث دور شدن ذرات از همگرایی نشود.

از طرفی دو متغیر تصادفی  $I_1$  و  $I_2$ ، در هر مرحله به صورت تصادفی استخراج می شوند، ممکن است شرایطی به وجود آید، که هر دو دارای مقدار بسیار بزرگ یا بسیار کوچک باشند. اگر هر دو متغیر دارای مقادیر بزرگ باشند، تجربیات بهترین پاسخ شخصی و بهترین پاسخ عمومی، به نوعی بیش از اندازه در رابطه سرعت دخیل می شوند و ذرات ممکن است از نقطه بهینه دور شوند. در حالت دوم، اگر هر دو متغیر دارای مقادیر بسیار کوچک باشند، در این حالت تجربیات شخصی و اجتماعی ذرات یا همان بهترین پاسخ شخصی و بهترین پاسخ عمومی، به اندازه کافی در رابطه سرعت دخیل نمی باشند و نقشی در به روز رسانی سرعت ندارند و همگرایی الگوریتم را کاهش می دهند.

اگر هر کدام از این مقادیر، به تنهایی نیز بسیار بزرگ یا بسیار کوچک باشند، مجدداً اثر مطلوبی روی سرعت نخواهند داشت، چون تاثیر یکی از عوامل بهترین پاسخ شخصی یا بهترین پاسخ عمومی را بیش از اندازه یا کمتر از مقدار مورد نیاز می کند. به عبارتی دیگر، دو متغیر وزن دهی تصادفی، تجربیات شخصی

و اجتماعی ذرات را بازتاب می دهند، اگر به صورت همزمان یکی بسیار کوچک و دیگری بسیار بزرگ باشد، ممکن است یکی از تجربیات شخصی یا عمومی بیش از اندازه بازتاب داده شود و دیگری نادیده گرفته شود و یا بالعکس، که باز هم این امر می تواند در سرعت ایجاد مشکل نماید و همگرایی الگوریتم را تحت تاثیر قرار دهد.

با در نظر گرفتن این عوامل و مدل کردن آن ها و استفاده از آن در به روز رسانی روابط، می توانیم روابط را به صورت رابطه های (۱-۵، ۲-۵، ۳-۵) بازنویسی کنیم.

$$v_a^i(k+1) = r_2 v_a^i + (1-r_2) c_1 r_1 (p_a^i - x_a^i(k)) + c_2 (1-r_2) (1 - r_1) (g_a^i - x_a^i(k)) \quad (1-5)$$

$$v_a^i(k+1) = \frac{v_a^i(k+1) \cdot v_a^{max}}{|v_a^i(k+1)|}, \text{ if } |v_a^i(k+1)| > v_a^{max} \quad (2-5)$$

$$x_a^i(k+1) = x_a^i(k) + v_a^i(k+1) \quad (3-5)$$

مقادیر  $r_1$  و  $r_2$ ، در این روابط، پارامترهای تصادفی هستند، که بین ۰ و ۱ انتخاب می شوند، همچنین حدود بیشینه و کمینه سرعت تعریف می شوند تا بتوانیم کنترل بیشتری روی سرعت حرکت ذرات داشته باشیم [۸۷].

### ۳.۵. ترکیب الگوریتم بهینه سازی توده ذرات بهینه شده و الگوریتم جستجوی

#### ممنوع

ضرایب شتاب دهنده  $c_1$  و  $c_2$ ، بیانگر شتابی هستند که به هر ذره داده می شوند تا به سمت محل بهترین شخصی و بهترین عمومی هدایت شوند. تنظیم این ضرایب مقدار تنش در سیستم را تغییر می دهد. مقدار



کم برای این ضرایب باعث می شود که ذرات قبل از کشیده شدن به سمت پاسخ به آرامی حرکت کنند، در حالی که مقدار زیاد منجر به حرکت سریع یا عبور از منطقه هدف می شود.

تجربه به دست آمده از اجرای الگوریتم نشان می دهد که می توان برای اکثر موارد مقدار این ثوابت را برابر با یکدیگر و برابر با ۲.۰ در نظر گرفت، اما در واقع، این انتخاب بستگی به نوع مسئله دارد. برای مسائل تک حالت، انتخاب  $c_2 > c_1$ ، حالت مناسب تری می باشد و برای مسائل چند حالتی این ضرایب به نحوی انتخاب می شوند، که  $c_2 < c_1$  باشد.

انتخاب صحیح و مناسب این ضرایب شتاب دهنده، نقش موثری در بهبود پاسخ های الگوریتم بهینه سازی توده ذرات دارد، لذا در الگوریتم پیشنهادی، بهینه سازی این ضرایب توسط الگوریتم جستجوی ممنوع پیاده سازی می شود.

تابع هدف الگوریتم جستجوی ممنوع، همان تابع هدف الگوریتم بهینه سازی توده ذرات می باشد. الگوریتم جستجوی ممنوع، یک نقطه نماینده پاسخ را در نظر می گیرد و با تعریف همسایگی روی فضای مسئله، به جستجو می پردازد. حرکت در جستجوی ممنوع به این نحو می باشد که، پس از یافتن بهترین پاسخ همسایگی حول آن تولید می شود، سپس اجزای این همسایگی در تابع هدف قرار می گیرند، پاسخ های به دست آمده با لیست ممنوع مورد بررسی قرار می گیرند تا اطمینان حاصل شود که این پاسخ ها قبلاً مشاهده نشده باشند. پاسخ های بررسی شده در صورتی که در لیست ممنوع فعال نباشند برای تکرار بعدی در لیست ممنوع قرار می گیرند و برای تعداد تکرارهای معین حالت ممنوع فعال می شوند تا از تکرار پاسخ ها و قرار گرفتن در حلقه جلوگیری شود. طول لیست ممنوع، بیانگر میزان حافظه استفاده شده الگوریتم می باشد.

الگوریتم معرفی شده برای آموزش شبکه عصبی توابع پایه شعاعی استفاده می شود. آموزش وزن های لایه میانی، عرض و مرکز توابع گاوسی توسط الگوریتم معرفی شده صورت می گیرد.

بدین منظور در ابتدا وزن های اولیه، مراکز و عرض های توابع گاوسی را به صورت تصادفی مقداردهی می کنیم. برای هر کدام از این موارد یک توده تولید می شود، در واقع یک توده برداری تعریف می کنیم که اندازه این بردار برابر با مجموع تعداد وزن ها، مراکز و عرض های توابع گاوسی می باشد.

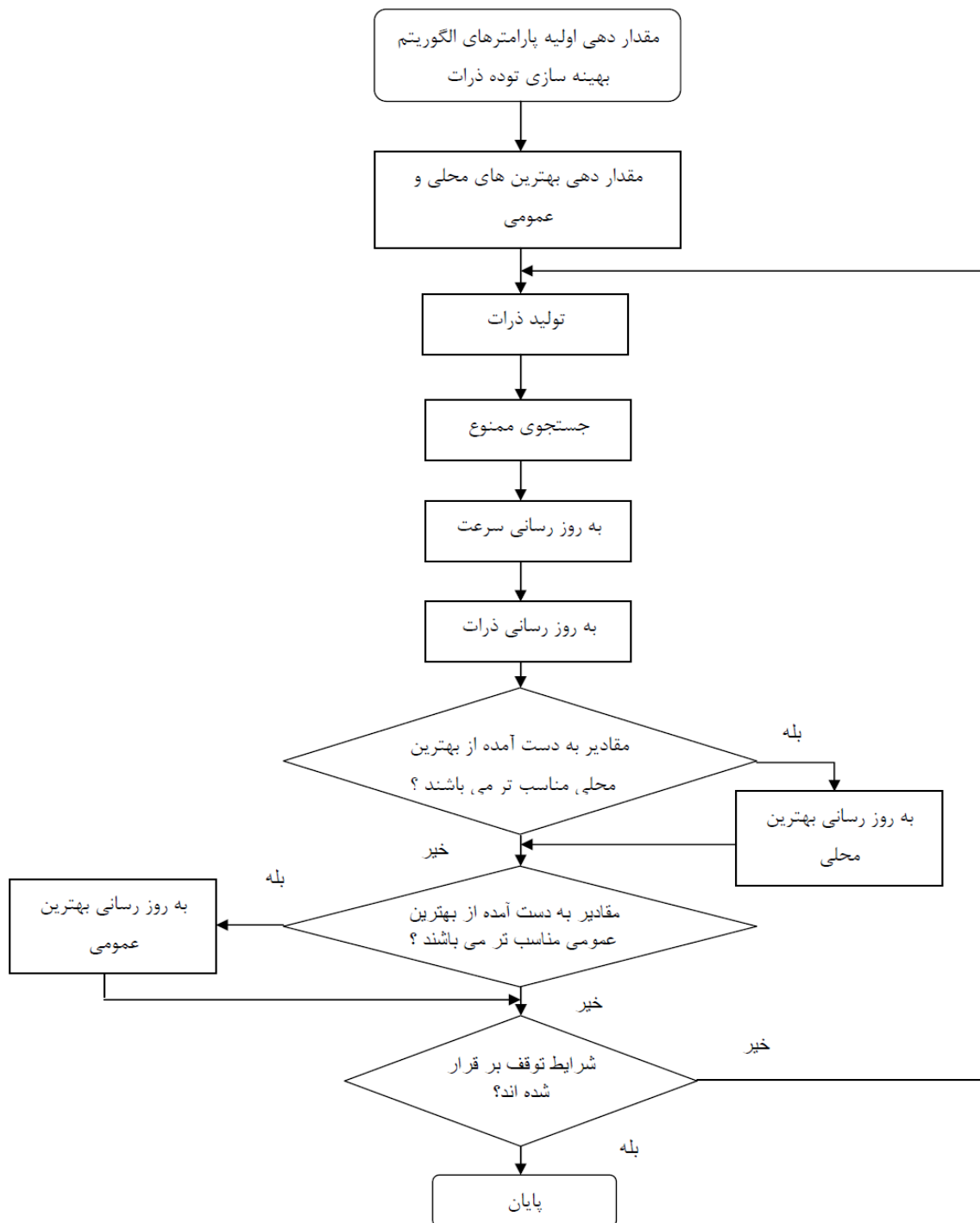
تابع هدف تابع خروجی شبکه عصبی می باشد، این تابع با توجه به مقادیر ورودی و وزن ها، محل مراکز و عرض های توابع گاوسی، خروجی هر ذره برداری از توده را محاسبه می کند و خطای آن را به عنوان خطا در نظر می گیرد. بردار یا ذره ای که کمترین خطا را داشته باشد به عنوان بهترین عمومی برگزیده می شود، و مقدار قبلی هر ذره یا بردار با مقدار جدید مقایسه می شود و در صورتی که موقعیت جدید نتیجه بهتری را در پی داشته باشد، این موقعیت به عنوان بهترین شخصی برای آن ذره یا بردار در نظر گرفته می شود.

با توجه به تجربیات شخصی و اجتماعی بدست آمده، حال باید به تولید سرعت هر ذره بپردازیم. برای این منظور، باید مقادیر مناسب  $C_1$  و  $C_2$ ، را به منظور به روزرسانی سرعت بهینه نماییم، در اینجا الگوریتم جستجوی ممنوع وارد می شود.

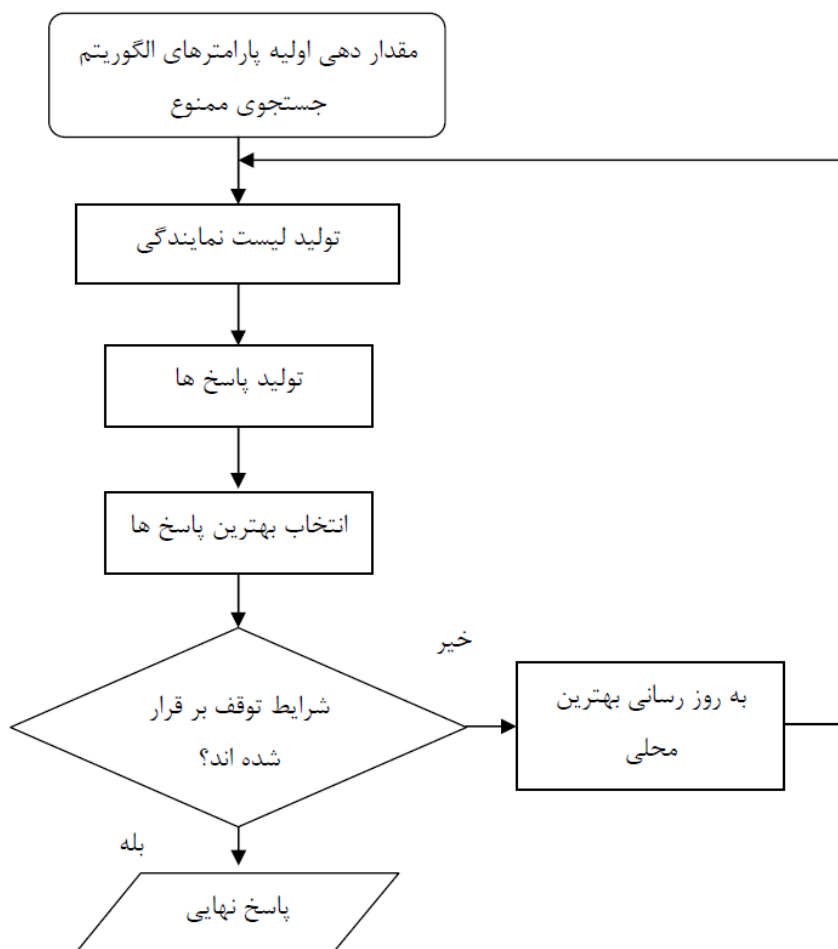
با توجه به نقطه شروع در نظر گرفته شده برای ضرایب شتاب دهنده، جستجو آغاز می شود. در ابتدا همسایگی مورد نظر برای این نقاط تولید می شود، سپس برای هر زوج شتاب دهنده سرعت عضو همسایگی، در ابتدا شرایط مورد بررسی قرار می گیرد که آیا این مقادیر در لیست ممنوع قرار دارند یا خیر، اگر در لیست ممنوع قرار داشته باشند از همسایگی حذف می شوند و همسایگی بعدی مورد بررسی قرار می گیرد، در غیر این صورت، با توجه به مقادیر ضرایب شتاب دهنده و تجربیات شخصی و اجتماعی

به دست آمده از توده ذرات، سرعت برای کلیه ذرات توده ایجاد می شود و با توجه به مقادیر به دست آمده برای سرعت، مقادیر موقعیت جدید را تولید کرده و با توجه به این مقادیر خروجی های الگوریتم را تولید می کنیم، این مقدار را با مقدار مطلوب مقایسه می کنیم و خطای مربوط به زوج مورد نظر را تولید می کنیم. سپس مقادیر زوج شتاب دهنده را در لیست ممنوع قرار داده و آن را برای دوره زمانی معینی ممنوع می کنیم تا از تکرار این نقاط جلوگیری شود. برای تمامی نقاط همسایگی، همین کار را انجام می دهیم، زوجی که کمترین خطا را داشته باشند، به عنوان مرکز همسایگی در تکرار بعدی الگوریتم جستجوی ممنوع انتخاب می شوند. فلوجارت الگوریتم پیشنهادی را در شکل های ۵-۱ و ۵-۲ مشاهده می کنیم.

در پایان عملکرد الگوریتم جستجوی ممنوع، مقادیر بهینه برای به روزرسانی سرعت های الگوریتم بهینه سازی توده ذرات تولید می شوند، حال سرعت های جدید تولید می شوند و با توجه به این مقادیر موقعیت های ذرات به روزرسانی می شوند و الگوریتم ادامه می یابد، در انتهای عملکرد الگوریتم پیشنهادی، مقادیر مطلوب وزن های لایه میانی، مراکز و عرض های توابع گاوسی به دست آمده اند و با توجه به این مقادیر خروجی های شبکه عصبی تولید می شوند و با مقایسه با مقادیر مطلوب خطاهای تکرار تولید می شوند، تا در تکرار های بعدی این خطا کاهش یابد.



شکل ۵-۲: فلوچارت الگوریتم پیشنهادی



شکل ۵-۳: فلوچارت زیربخش جستجوی ممنوع از الگوریتم پیشنهادی

شبه کد الگوریتم پیشنهادی برای آموزش شبکه عصبی در زیر آورده شده است:

- مقدار دهی اولیه پارامترهای شبکه عصبی
- مقدار دهی اولیه پارامترهای الگوریتم بهینه سازی توده ذرات
- مقدار دهی اولیه پارامترهای الگوریتم جستجوی ممنوع
- اعمال ورودی به شبکه

- برای تعداد تکرارهای شبکه عصبی
  - مقادیر ورودی لایه میانی تولید شوند
  - برای تعداد تکرار های الگوریتم بهینه سازی توده ذرات
    - موقعیت ها به روز رسانی شوند
    - تابع هدف تولید شود و خطاها استخراج شود
    - بهترین تجربیات شخصی و عمومی به روز رسانی شوند
    - برای تعداد تکرار های الگوریتم جستجوی ممنوع
- همسایگی جدید تولید شود
- همسایگی با لیست ممنوع بررسی شود و نقاط نامطلوب از همسایگی حذف شوند
- مقادیر تابع هدف و خطاها برای کل همسایگی تولید شود
- مرکز جدید همسایگی انتخاب شود
- لیست ممنوع به روز رسانی شود
- پایان
- سرعت ها به روز رسانی شوند
- مقادیر تابع هدف و خطاها برای کل توده تولید شود
- پایان
- مقادیر تابع هدف و خطاها برای ورودی ها تولید شوند
- پایان
- پایان

## ۴.۵. نتایج حاصل از الگوریتم پیشنهادی

نتایج حاصل از آموزش شبکه عصبی توابع پایه شعاعی با استفاده از الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع به صورت زیر می باشد، نقاط، روی نمودار، مقادیر اصلی تابع می باشند و خط ممتد تقریبی است که روی تابع صورت گرفته است، مقادیر به صورت مرتب شده از کم به زیاد می باشند تا میزان تطبیق را به درستی مشاهده کنیم.

به منظور بررسی نتایج حاصل از الگوریتم پیشنهادی، از ۲۰ تابع استاندارد مختلف استفاده شده است. لیست این توابع در ضمیمه ۱ آورده شده است. داده های متغیرهای ورودی بین ۱- تا ۱ نرمال شده اند. تعداد تکرارهای شبکه عصبی برابر با ۱۰ در نظر گرفته شده و تعداد تکرارهای الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع ۳۰ در نظر گرفته شده است. طول همسایگی برای قسمت جستجوی ممنوع برابر با ۵ در نظر گرفته شده است. دوره تناوب ممنوع برای حالت ممنوع برابر با ۵ در نظر گرفته شده است و اندازه توده ذرات با توجه به جدول زیر برابر با ۶۴ در نظر گرفته شده است. این انتخاب به منظور ایجاد یک مصالحه بین زمان محاسبه و نرخ همگرایی می باشد. اندازه ابعاد توده ذرات با توجه به پیچیدگی مساله عمدتاً بین ۲۰ تا ۱۰۰ انتخاب می شود، اما در مساله آموزش شبکه عصبی با انتخاب اندازه توده ذرات برای مقادیر کمتر از ۲۵، واگرایی خواهیم داشت. همچنین به طور کلی با افزایش ابعاد توده ذرات، نرخ همگرایی افزایش می یابد و این به دلیل بیشتر شدن گزینه های پاسخ می باشد که احتمال یافتن پاسخ مناسب تر را افزایش می دهد. اما با انتخاب مقادیر بزرگتر، باید بهای پردازش و حجم محاسبات بیشتر را نیز بپردازیم. در جدول (۵-۱) مقایسه ای بین زمان پردازش الگوریتم برای تعداد توده ذرات مختلف آورده شده است.

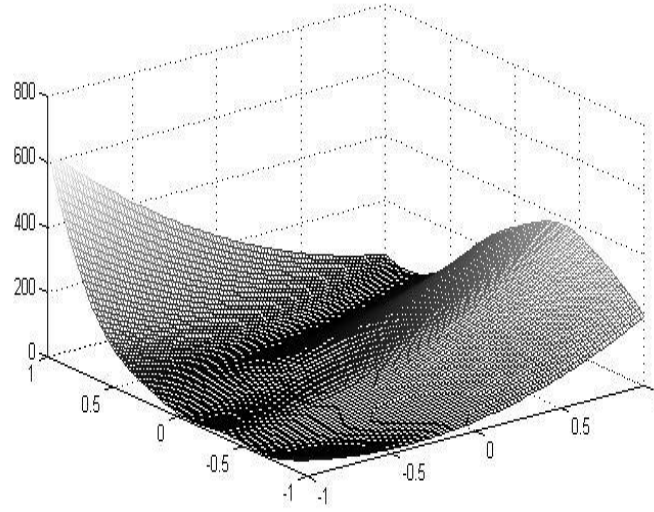
جدول ۵-۱: بررسی زمان تقریبی آموزش با توجه به اندازه ابعاد توده ذرات

	۲۵	۳۰	۳۶	۶۴	۱۰۰
زمان تقریبی محاسبه (ثانیه)	۶۹	۸۰	۹۵	۲۳۲	۳۸۵

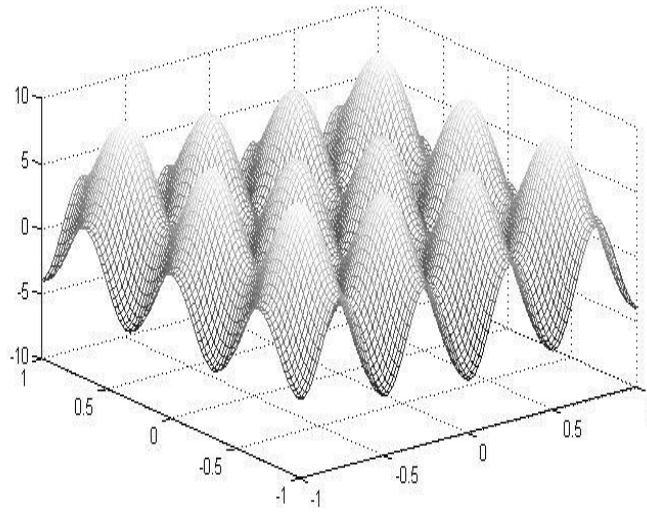
از آنجایی که توابع دارای ورودیهایی با تعداد متغیر ۲، ۳ و یا بیشتر می باشند، لذا محور افقی را شماره ورودی در نظر گرفتیم، ازین رو، محور افقی به جای مقدار محدوده متغیر ورودی، دارای درجه بندی از ۱ تا ۱۰۰، (برابر با تعداد ورودی های تست)، می باشد.

همچنین از آنجایی که توابع دارای پیچیدگی زیاد می باشند، که این مساله در شکل های (۳-۵)، (۴-۵) و (۵-۵) برای ۳ نمونه از توابع، به وضوح قابل رویت است (اکثر توابع دارای تعداد ورودی بیشتر از ۲ می باشند، لذا فقط توابعی که دارای ۲ ورودی می باشند به صورت ۳ بعدی به عنوان نمونه رسم شده اند)، با رسم عادی نمودارهای خروجی و تقریب نمی توان به درستی تطبیق را مشاهده نمود، به همین دلیل، به منظور بررسی بهتر تطبیق، منظم کردن داده های تست به نحوی صورت گرفته تا مقادیر خروجی به صورت صعودی باشند، به همین دلیل است که اکثر نمودارها شکل به ظاهر یکسانی دارند، اما اگر به درستی بنگریم، مقادیر دامنه خروجی متفاوتی در نمودارها مشاهده می کنیم.

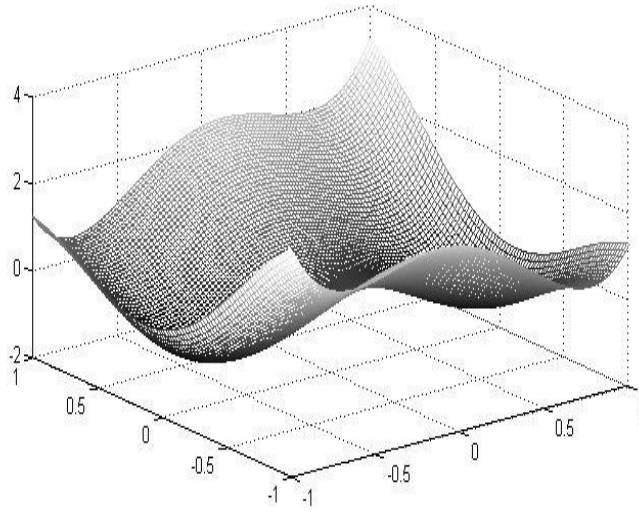




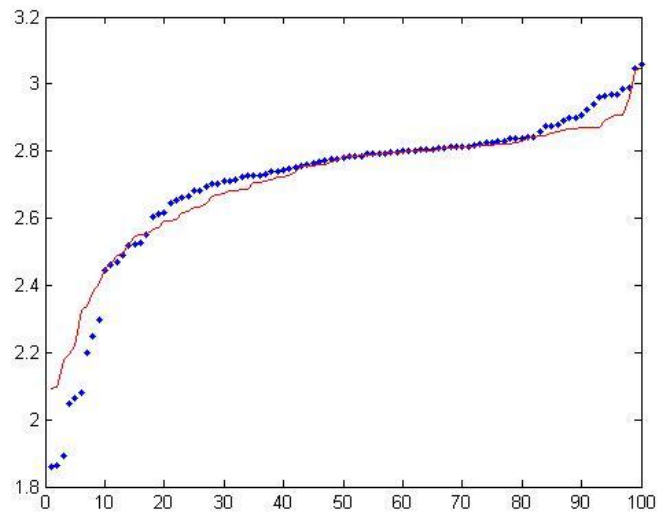
شکل ۵-۴: رسم ۳ بعدی تابع RC



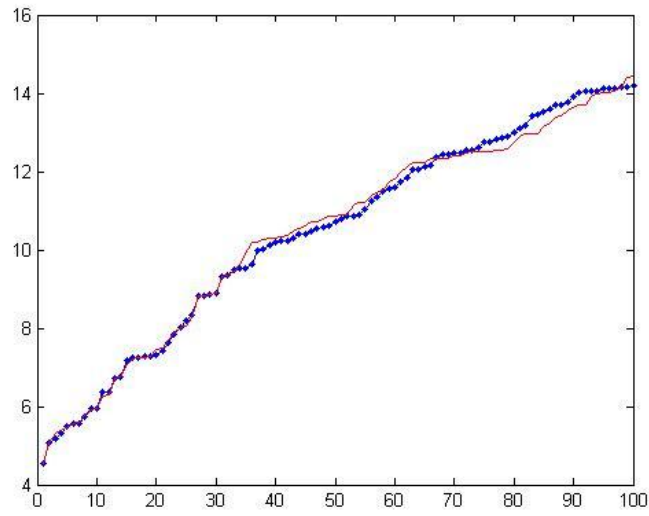
شکل ۵-۵: رسم ۳ بعدی تابع B



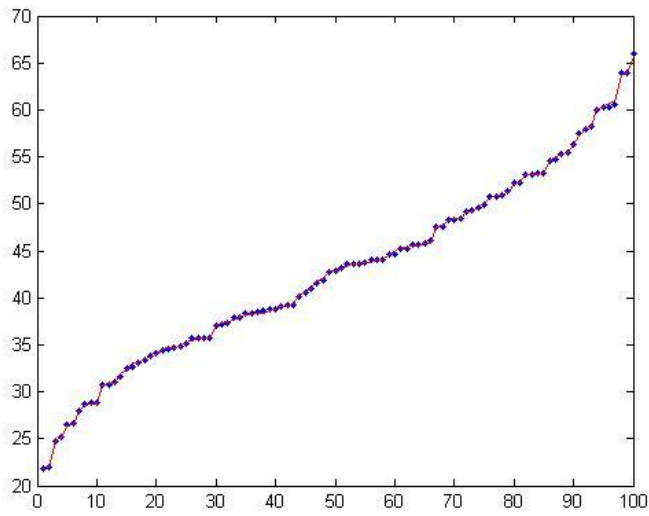
شکل ۵-۶: رسم ۳ بعدی تابع H



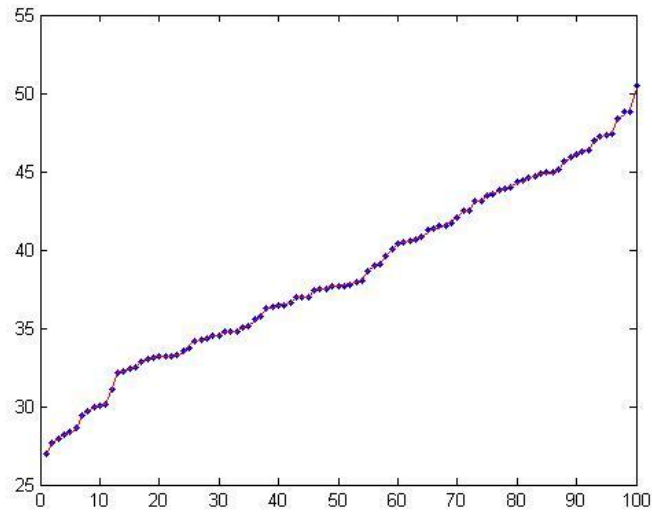
شکل ۵-۷: تقریب تابع AK با استفاده از الگوریتم پیشنهادی



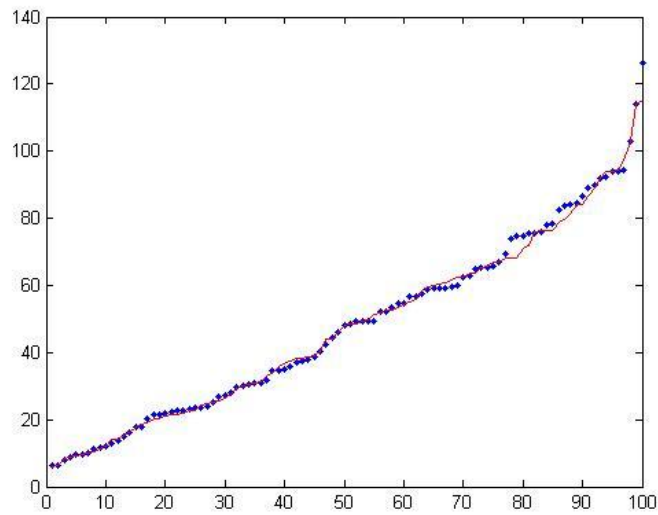
شکل ۵-۸: تقریب تابع BL با استفاده از الگوریتم پیشنهادی



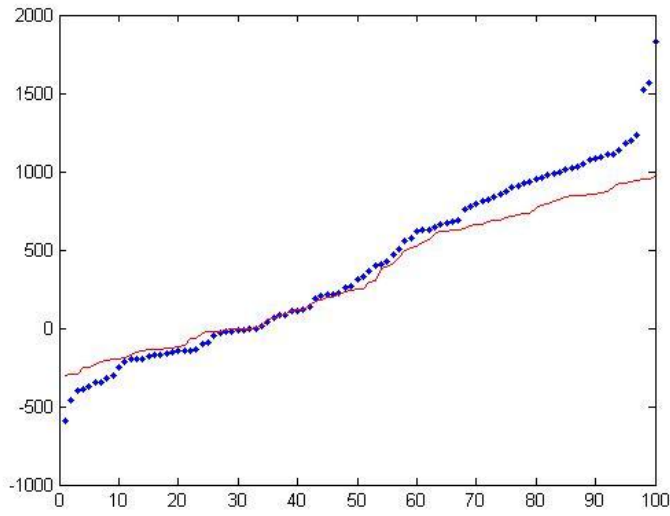
شکل ۵-۹: تقریب تابع BO با استفاده از الگوریتم پیشنهادی



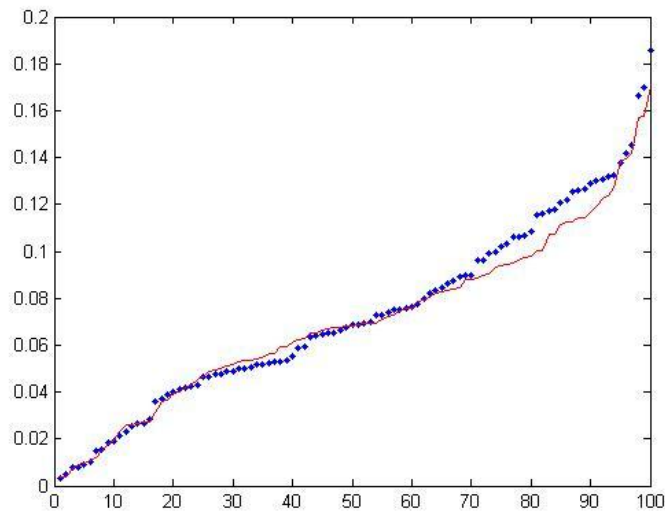
شکل ۵-۱۰: تقریب تابع RC با استفاده از الگوریتم پیشنهادی



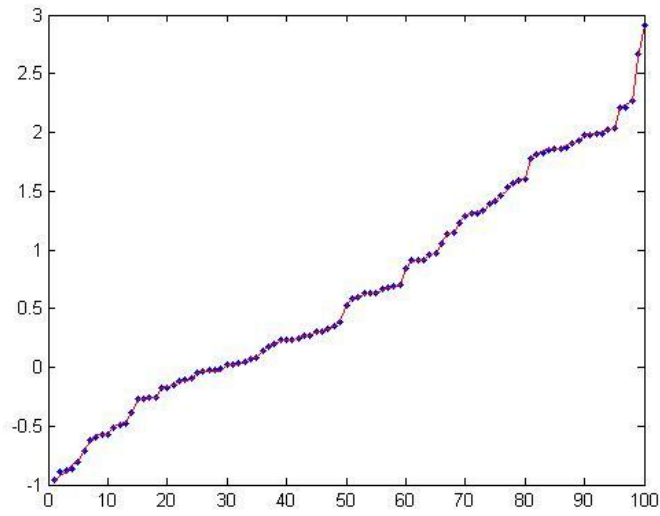
شکل ۵-۱۱: تقریب تابع CV با استفاده از الگوریتم پیشنهادی



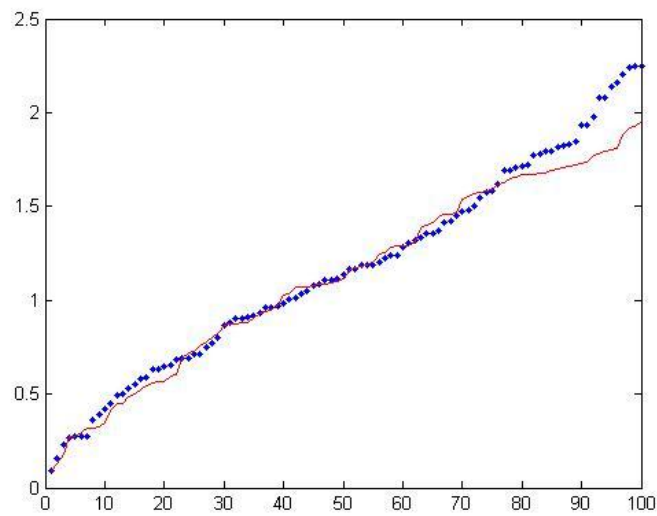
شکل ۵-۱۲: تقریب تابع GP با استفاده از الگوریتم پیشنهادی



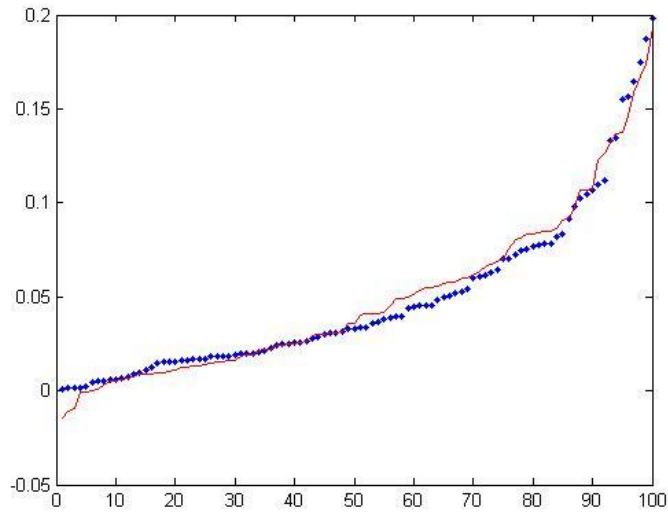
شکل ۵-۱۳: تقریب تابع G با استفاده از الگوریتم پیشنهادی



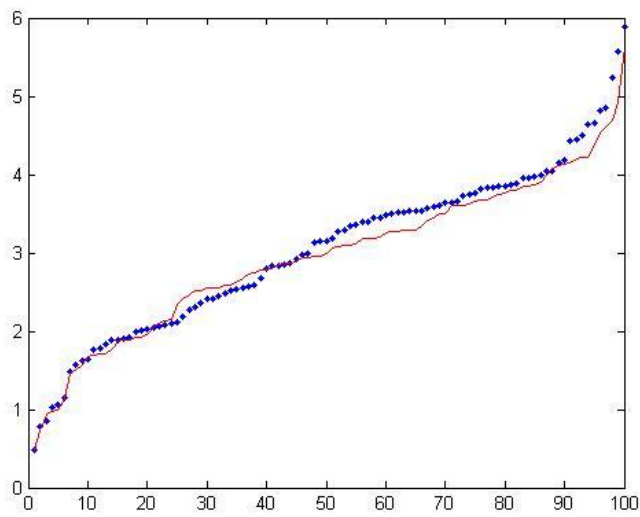
شکل ۵-۱۴: تقریب تابع HM با استفاده از الگوریتم پیشنهادی



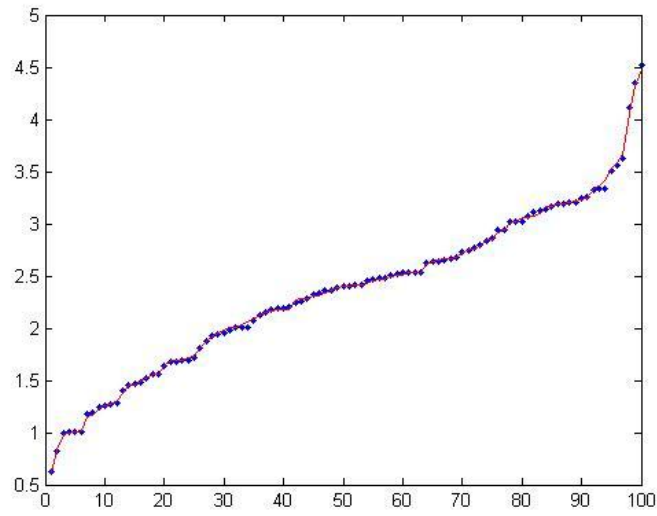
شکل ۵-۱۵: تقریب تابع L با استفاده از الگوریتم پیشنهادی



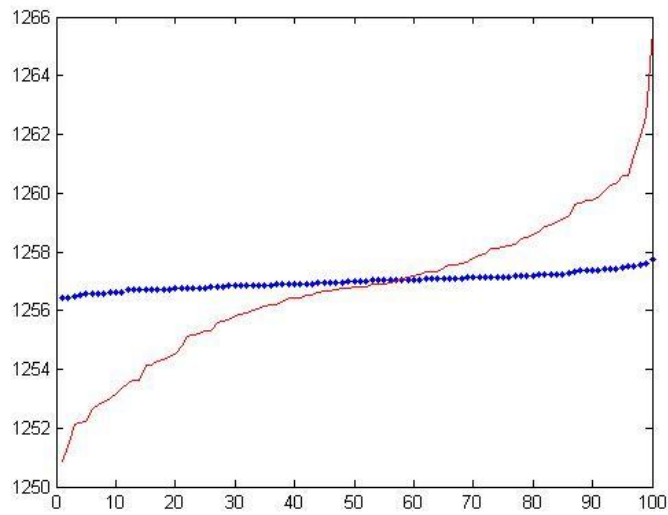
شکل ۵-۱۶: تقریب تابع MT با استفاده از الگوریتم پیشنهادی



شکل ۵-۱۷: تقریب تابع RT با استفاده از الگوریتم پیشنهادی

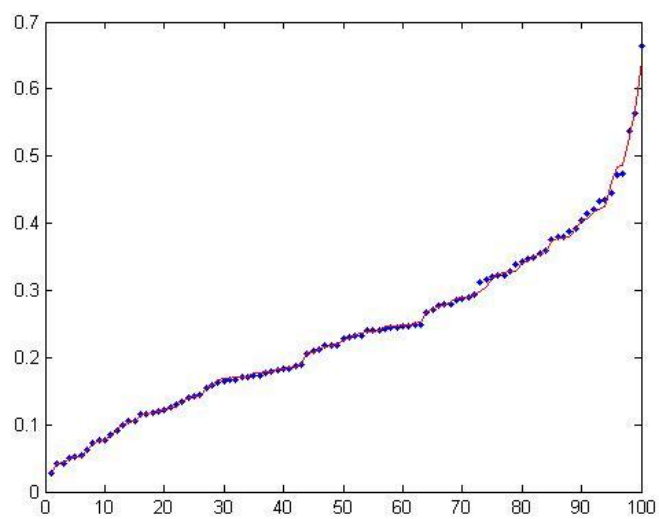


شکل ۵-۱۸: تقریب تابع R با استفاده از الگوریتم پیشنهادی

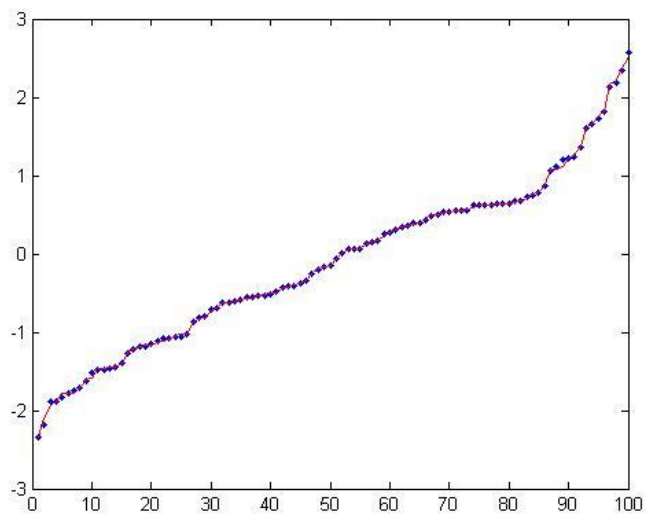


شکل ۵-۱۹: تقریب تابع SC با استفاده از الگوریتم پیشنهادی

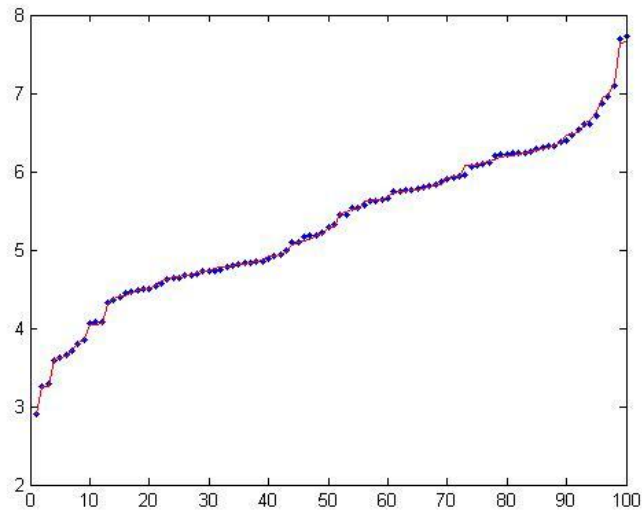




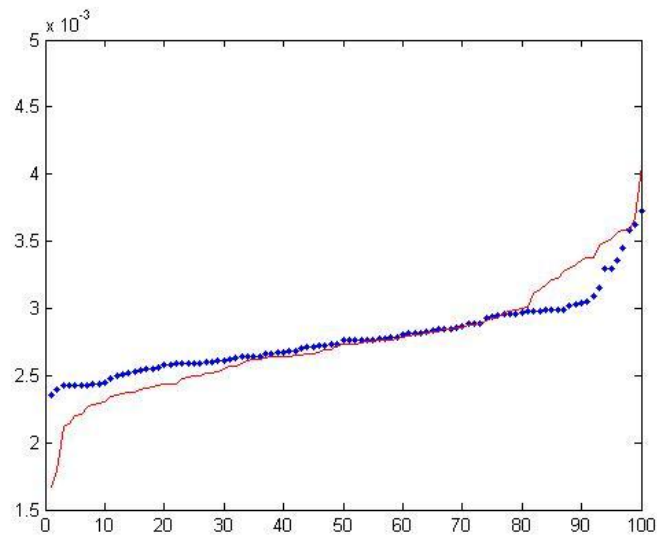
شکل ۵-۲۰: تقریب تابع SR با استفاده از الگوریتم پیشنهادی



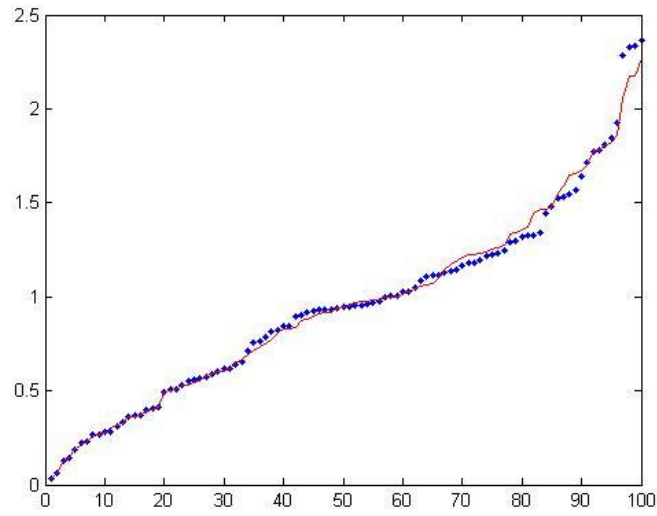
شکل ۵-۲۱: تقریب تابع SS با استفاده از الگوریتم پیشنهادی



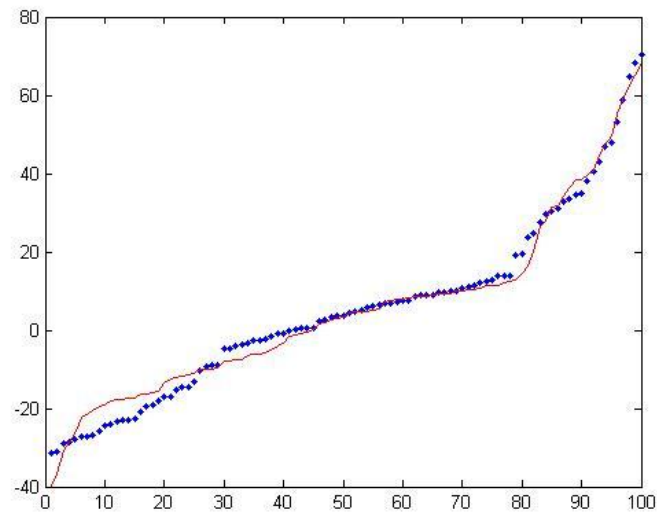
شکل ۵-۲۲: تقریب تابع  $T$  با استفاده از الگوریتم پیشنهادی



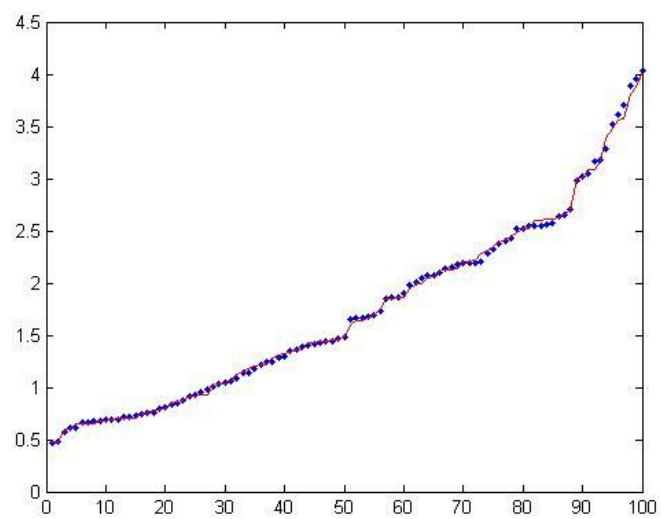
شکل ۵-۲۳: تقریب تابع  $Z$  با استفاده از الگوریتم پیشنهادی



شکل ۵-۲۴: تقریب تابع DJ با استفاده از الگوریتم پیشنهادی



شکل ۵-۲۵: تقریب تابع B با استفاده از الگوریتم پیشنهادی



شکل ۵-۲۶: تقریب تابع DP با استفاده از الگوریتم پیشنهادی

## ۶. بررسی و مقایسه نتایج

الگوریتم پیشنهادی با استفاده از نرم افزار متلب نسخه ۲۰۱۰b portable پیاده سازی شده است.

به منظور بررسی نتایج حاصل از الگوریتم پیشنهادی، از ۲۰ تابع استاندارد مختلف استفاده شده است. لیست این توابع در ضمیمه ۱ آورده شده است. داده های متغیرهای ورودی بین ۱- تا ۱ نرمال شده اند. توابع دارای تعداد متغیرهای مختلف می باشند. تعداد تکرارهای شبکه عصبی برابر با ۱۰ در نظر گرفته شده و تعداد تکرارهای الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع ۳۰ در نظر گرفته شده است. طول همسایگی برای قسمت جستجوی ممنوع برابر با ۵ در نظر گرفته شده است. دوره تناوب ممنوع برای حالت ممنوع برابر با ۵ در نظر گرفته شده است و اندازه توده ذرات برابر با ۶۴ می باشد.

انتخاب اندازه توده ذرات با توجه به چند نمونه تست الگوریتم و به صورت تجربی استفاده شده است، در صورت انتخاب مقدار کمتر همگرایی الگوریتم دچار مشکل شده و همگرایی مناسبی نخواهد داشت و در صورت انتخاب مقادیر بیشتر برای تعداد توده، تاثیر مثبتی در میزان همگرایی نخواهیم داشت و صرفاً حجم محاسبات و زمان اجرای الگوریتم افزایش خواهد یافت.

تعداد نرون های میانی برابر با تعداد ورودیها، مقادیر وزن ها، مراکز و عرض های اولیه به صورت تصادفی ایجاد می شوند، ضرایب شتاب دهنده نیز توسط الگوریتم جستجوی ممنوع بهینه سازی می شوند، و به روز رسانی وزن ها، مراکز و عرض های توابع گاوسی توسط الگوریتم مرکب بهینه سازی توده ذرات و جستجوی ممنوع صورت می گیرد.

نتایج الگوریتم پیشنهادی، با نتایج حاصل از شبکه عصبی توابع پایه شعاعی با آموزش به روش گرادیان نزولی و آموزش به روش بهینه سازی توده ذرات ساده مقایسه شده است، تا مزایا و معایب آن مورد بررسی قرار گیرد.

شبه کد آموزش شبکه عصبی به روش گرادیان نزولی در زیر آورده شده است:

- مقداردهی اولیه پارامترهای شبکه عصبی
- اعمال ورودی به شبکه
- برای تعداد تکرارهای شبکه
  - مقادیر ورودی لایه میانی تولید شوند
  - مقادیر تابع هدف تولید و خطاها تولید شود
  - با توجه به خطای به دست آمده مقادیر وزن ها، مراکز و عرض ها توسط روابط (۶-۱، ۶-۲، ۶-۳) به روز رسانی شوند.
  - پایان
- پایان

شبه کد آموزش شبکه عصبی به روش الگوریتم بهینه سازی توده ذرات در زیر آورده شده است:

- مقدار دهی اولیه پارامتر های شبکه عصبی
- مقدار دهی اولیه پارامتر های الگوریتم بهینه سازی توده ذرات
- مقدار دهی اولیه پارامتر های الگوریتم جستجوی ممنوع
- اعمال ورودی به شبکه
- برای تعداد تکرارهای شبکه عصبی
  - مقادیر ورودی لایه میانی تولید شوند
  - برای تعداد تکرار های الگوریتم بهینه سازی توده ذرات
    - موقعیت ها به روز رسانی شوند
    - تابع هدف تولید شود و خطاها استخراج شود
    - بهترین تجربیات شخصی و عمومی به روز رسانی شوند

- سرعت ها به روز رسانی شوند
- مقادیر تابع هدف و خطاها برای کل توده تولید شود
- پایان

○ مقادیر تابع هدف و خطاها برای ورودی ها تولید شوند

○ پایان

• پایان

در آموزش شبکه عصبی با روش گرادیان نزولی، تعداد تکرارها، ۵۰۰ تکرار می باشد، تعداد نرون های میانی برابر با تعداد ورودیها، مقادیر اولیه وزن ها، مراکز و عرض ها به صورت تصادفی ایجاد می شوند و روابط به روز رسانی وزن ها، مراکز و عرض های توابع گاوسی به صورت روابط (۱-۶)، (۲-۶)، (۳-۶) می باشد.

$$w(n+1) = w(n) - \eta_r \frac{\partial E}{\partial w} \quad (1-6)$$

$$\mu(n+1) = \mu(n) - \eta_r \frac{\partial E}{\partial \mu} \quad (2-6)$$

$$\sigma^2(n+1) = \sigma^2(n) - \eta_1 \frac{\partial E}{\partial \sigma^2} \quad (3-6)$$

روابط پس از استخراج روابط گرادیان به صورت روابط (۴-۶)، (۵-۶)، (۶-۶) خواهند بود.

$$y_m = \exp\left(-\frac{\|x - v_m\|^2}{2\sigma_m^2}\right) \quad (4-6)$$

$$s_j = \sum_m y_m w_{mj} \quad (5-6)$$

$$z_j = \frac{1}{1 + e^{-s_j}} \quad (6-6)$$

X: بردار ورودی



V: بردار مرکز توابع پایه شعاعی

σ: بردار عرض توابع پایه شعاعی

Y: بردار مقادیر خروجی لایه مخفی

W: بردار وزن نرون های لایه مخفی

Z: بردار خروجی

در آموزش شبکه عصبی به کمک الگوریتم بهینه سازی توده ذرات، تعداد تکرارهای شبکه ۵۰ تکرار در نظر گرفته شده است. همچنین تعداد تکرار های الگوریتم بهینه سازی توده ذرات ۳۰ تکرار می باشد. اندازه توده ذرات ۶۴ در نظر گرفته شده است. ضرایب شتاب دهنده  $C_1$  و  $C_2$  مساوی با یکدیگر و برابر با مقدار ثابت ۲ در نظر گرفته شده اند. تعداد نرون های میانی برابر با تعداد ورودیها، وزن ها، مراکز و عرض های اولیه به صورت تصادفی ایجاد می شوند و به روز رسانی آن ها توسط الگوریتم بهینه سازی توده ذرات صورت می گیرد.

مجدداً انتخاب پارامترها به صورت تجربی، به منظور دستیابی به بهترین نتیجه و همگرایی می باشد. انتخاب یکسان ضرایب اصلاح سرعت، به این دلیل است که در الگوریتم کلاسیک برای حل اکثر مسائل به صورت یکسان و برابر با مقدار ۲ در نظر گرفته می شود، و همچنین به منظور مقایسه این انتخاب با بهینه سازی این ضرایب و بررسی اثر این موضوع این کار صورت پذیرفته است.

در نهایت الگوریتم پیشنهادی مرکب بهینه سازی توده ذرات و جستجوی ممنوع با روش گرادیان نزولی و الگوریتم بهینه سازی توده ذرات برای آموزش شبکه عصبی از نظر میزان خطای نهایی و سرعت همگرایی با هم مورد مقایسه قرار می گیرند.

جدول ۶-۱: خطا های نهایی حاصل از روش های مختلف برای آموزش شبکه عصبی برای توابع مورد بررسی

$f$	GD	PSO	AMPSO-TS
AK	۴.۹۶۱۹	۳.۷۸۸۰	۳.۴۵۷۶
BL	۱.۹۱۶۰	۰.۵۸۷۲	۰.۴۶۷۷
L	۲.۷۵۶۹	۱.۱۳۴۲	۰.۹۱۲۰
MT	۸۷.۵۱۲۸	۱۵.۰۶۶۷	۱۰.۰۰۹۲
BO	۰.۹۷۷۹	۰.۶۷۹۳	۰.۴۵۱۶
RT	۱۱.۶۲۹۵	۴.۷۵۶۱	۲.۶۱۰۳
RC	۴.۰۱۵۸	۰.۴۱۳۸	۰.۳۸۱۰
R	۴.۰۸۱۶	۱.۰۰۲۳	۰.۷۷۸۱
CV	۱۰.۶۰۰۳	۹.۵۳۸۱	۹.۵۰۱۴
SC	۰.۵۶۸۷	۰.۵۵۳۴	۰.۱۰۸۸
DJ	۳.۳۶۵۵	۲.۱۳۱۳	۲.۰۵۰۴
SR	۱۳.۶۳۹۹	۴.۴۱۹۰	۳.۲۶۷۱
DP	۱۳.۷۶۵۴	۵.۲۱۶۳	۲.۲۹۳۷
SS	۱۱.۳۵۴۴	۶.۶۸۵۱	۵.۶۴۰۴
GP	۲۹.۴۳۶۶	۷.۴۲۱۱	۲.۸۱۰۶
T	۱۲.۰۹۹۳	۸.۷۲۳۵	۵.۹۴۲۰
G	۴۷.۰۷۲۲	۴.۹۰۰۳	۰.۸۴۲۶

Z	۰.۳۱۲۷	۰.۲۹۲۵	۰.۰۹۱۳
HM	۳۸.۲۰۸۶	۳۲.۰۱۲۹	۲۹.۴۷۶۸
B	۱۱۷.۰۴۰۴	۷۸.۴۵۳۸	۶۲.۱۲۶۷

جدول ۶-۲: میزان همگرایی روش های بهینه سازی توده ذرات و الگوریتم پیشنهادی در طول آموزش

$f$	PSO (تعداد تکرارها)			AMPSO-TS (تعداد تکرارها)		
	۱۰	۲۰	۳۰	۵	۱۰	۱۵
AK	۴.۸۸۷۶	۴.۴۲۴۱	۴.۱۵۰۳	۳.۴۵۷۶	۳.۴۵۷۶	۳.۴۵۷۶
BL	۰.۶۹۵۴	۰.۶۷۶۳	۰.۵۸۷۲	۰.۴۶۷۷	۰.۴۶۷۷	۰.۴۶۷۷
BO	۰.۸۰۸۴	۰.۷۶۷۸	۰.۶۷۹۳	۰.۵۰۴۷	۰.۴۶۳۳	۰.۴۵۷۲
RC	۰.۸۹۱۰	۰.۶۷۴۰	۰.۴۲۰۳	۰.۸۵۴۶	۰.۷۷۲۷	۰.۵۸۱۰
CV	۱۰۰.۰۸۱۰	۹.۶۲۸۸	۹.۵۴۸۷	۹.۸۲۳۶	۹.۵۰۱۴	۹.۵۰۱۴
DJ	۴.۲۲۵۵	۲.۷۷۸۶	۲.۶۶۰۸	۲.۹۳۰۳	۲.۴۷۳۷	۲.۲۵۰۴
DP	۶.۵۷۵۷	۶.۴۶۵۹	۵.۲۹۲۶	۴.۳۱۱۴	۳.۱۸۲۹	۲.۲۹۳۷
Z	۰.۸۶۵۴	۰.۶۱۸۵	۰.۴۲۰۴	۰.۳۰۰۵	۰.۱۹۷۷	۰.۱۴۲۸

نتایج حاصل از مشاهدات فوق بیانگر این مطلب می باشد که انتخاب بهینه ضرایب  $C_1$  و  $C_2$  ، می تواند تاثیر بسزایی در نرخ همگرایی توده ذرات در الگوریتم بهینه سازی داشته باشد. همانگونه که مشاهده می

شود در الگوریتم پیشنهادی، با تعداد تکرار های کمتری نسبت به الگوریتم اصلی بهینه سازی توده ذرات به پاسخی با دقتی بهتر از الگوریتم اصلی دست می یابیم. این امر تاثیر انتخاب بهینه ضرایب  $C_1$  و  $C_2$  را در میزان نرخ همگرایی الگوریتم را تأیید می کند، زیرا همانگونه که مشاهده می شود در اکثر موارد در الگوریتم پیشنهادی پس از ۵ تکرار به دقتی بالا تر از ۳۰ تکرار الگوریتم بهینه سازی توده ذرات دست یافتیم که این امر میزان همگرایی بالای این الگوریتم را نشان می دهد.

با توجه به نتایج به دست آمده انتخاب مقدار ثابت برای ضرایب  $C_1$  و  $C_2$ ، تا حد قابل قبولی برای رسیدن به پاسخ مناسب می باشد، اما زمانی که به دقت بالاتر و مناسب تری نیاز داشته باشیم باید این ضرایب را نیز به نحو بهینه انتخاب کنیم.

در انتها نیز مقادیر بهینه سازی شده ضرایب  $C_1$  و  $C_2$ ، را در الگوریتم پیشنهادی در جدول زیر مشاهده می کنیم.

جدول ۳-۶: مقدار بهینه سازی شده ضرایب C<sub>1</sub> و C<sub>2</sub> برای آموزش توابع مختلف

	C <sub>1</sub>	C <sub>2</sub>
AK	۱.۷۴۳۶	۱.۷۴۳۶
RC	۰.۱۰۲۶	۳.۲۸۲۱
B	۱.۷۴۳۶	۰.۴۱۰۳
BL	۱.۷۴۳۶	۱.۷۴۳۶
BO	۱.۱۲۸۲	۲.۷۶۹۲
CV	۰.۴۱۰۳	۳.۴۸۷۲
DJ	۱.۷۴۳۶	۱.۷۴۳۶
DP	۲.۵۶۴۱	۱.۱۲۸۲
GP	۱.۳۳۳۳	۲.۴۶۱۵
G	۲.۶۶۶۷	۰.۵۱۲۸
HM	۰.۱۰۲۶	۳.۱۷۹۵
L	۰.۳۰۷۷	۲.۲۵۶۴
MT	۰.۲۰۵۱	۱.۸۴۶۲
RT	۰.۴۱۰۳	۳.۱۷۹۵
R	۰.۹۲۳۱	۲.۳۵۹۰
SC	۱.۰۲۵۶	۲.۰۵۱۳
SR	۱.۹۴۸۷	۱.۸۴۶۲
SS	۰.۹۲۳۱	۲.۰۵۱۳

T	۰.۲۰۵۱	۳.۶۹۲۳
Z	۲.۱۵۳۸	۱.۷۴۳۶

همانگونه که مشاهده می شود، مقادیر ضرایب  $c_1$  و  $c_2$  ، با مقدار ثابت ۲، در اکثر موارد تفاوت بسیار زیادی دارد، در نتیجه با ثابت در نظر گرفتن این ضرایب، نرخ همگرایی کاهش می یابد. بنابراین به منظور دستیابی به دقت بالاتر و دستیابی به نرخ همگرایی بالاتر نیاز به انتخاب درست ضرایب اصلاح داریم، که این امر نیز بر لزوم بهینه سازی این ضرایب اصلاح صحنه می گذارد.

## ۷. پیشنهادات و کارهای آینده

## ۱.۷. پیشنهادات و کارهای آینده

مشاهده شد که انتخاب مختلف ضرایب اصلاح تا چه حد می تواند روی دقت پاسخ و میزان همگرایی الگوریتم تاثیرگذار باشد، همچنین نحوه انتخاب این ضرایب می تواند به روش های مختلفی صورت گیرد، در اینجا از الگوریتم جستجوی ممنوع استفاده شده است که پاسخ های مناسبی را در پی دارد اما می توان این کار را با سایر الگوریتم های بهینه سازی مانند تبرید شبیه سازی شده، کلونی مورچگان و یا الگوریتم های بهینه سازی دیگری نیز مورد بررسی قرار داد، تا مقایسه ای روی دقت نتایج به دست آمده و زمان اجرای الگوریتم و یا حجم محاسبات انجام شده صورت داد، که این طرح به منظور کار آتی اینجانب و طرح مقاله ای مناسب در حال بررسی و اجرا می باشد.

همچنین از الگوریتم معرفی شده می توان در کاربرد های گوناگونی بهره گرفت، یکی از این کاربردها بهینه سازی در یافتن نقاط بهینه یا کمینه ی توابع می باشد، که در این کار نیز مراحل مطالعاتی را پشت سر گذاشته و مراحل پیاده سازی آن در دست اجرا می باشد.

در حال حاضر تحقیقات اینجانب بر روی الگوریتم های بهینه سازی و کاربرد های مختلف آن ها ادامه دارد و استفاده از این الگوریتم به صورت مجزا و در ترکیب با یکدیگر بر روی کاربرد های مختلف را در دست پیاده سازی دارم. چندین مورد از کارهای ذکر شده مراحل مقدماتی را پشت سر گذاشته و مراحل نهایی طرح مقاله را سپری می کند و در نهایت نتایج بررسی ها و تحقیقات به صورت یک مطالعه کلی روی الگوریتم ها در قالب یک مقاله مطالعاتی به چاپ خواهد رسید.

اما به عنوان پیشنهادی برای کارهای آتی در این زمینه می توان به این مساله اشاره کرد که، الگوریتم های جستجو و بهینه سازی جدیدی نیز در سال های اخیر معرفی شده اند که جای کار بسیار زیادی روی این الگوریتم ها وجود دارد، از این جمله الگوریتم ها می توان به الگوریتم های رقابت استعماری و الگوریتم



فاخته اشاره نمود که مدت زمان بسیار کوتاهی از معرفی آن ها می گذرد، و می توان به عنوان یک طرح پیشنهادی، از این الگوریتم ها نیز در آموزش شبکه های عصبی استفاده نمود.

به عنوان ایده ای دیگر نیز می توان موضوع پیاده سازی شده در این پایان نامه را، به جای استفاده از شبکه های عصبی توابع پایه شعاعی، از شبکه های عصبی موجک پیاده سازی نمود. همانگونه که در متن پایان نامه بیان شد، این نوع از شبکه های عصبی نیز در زمینه تقریب توابع از بازدهی مناسبی برخوردار می باشند.

در حال حاضر از روی خلاصه مطالب بیان شده در این پایان نامه مقاله ای تنظیم شده و به چاپ رسیده است، مشخصات مقاله فوق و مقاله دیگری که کارهای مطالعاتی آن به پایان رسیده است و در مرحله پیاده سازی بر روی بهبود آن مشغول به کار می باشم نیز در ادامه آورده شده است.

## ۲.۷. لیست مقالات استخراج شده

### چاپ شده

Training RBF Neural Network by hybrid Adaptive Modified Particle Swarm Optimization – Tabu Search Algorithm for Function Approximation, IJSER, August ۲۰۱۲, Vol ۳, Issue ۸

### در دست تدوین

Global optimization by hybrid Adaptive Modified Particle Swarm Optimization – Tabu Search Algorithm

۸. ضمایم

جدول ۸-۱: روابط توابع مورد بررسی

$f$	Function Name	Formula
AK	Ackley	$AK_n(x) = 20 + e - 20 \cdot e^{-\frac{1}{\sqrt{n}} \sum_{i=1}^n x_i^2}$ $- e^{-\frac{1}{n} \sum_{i=1}^n \cos(\pi x_i)}$
RC	Brain Rcos	$RC(x) = (x_2 - \frac{\Delta}{4\pi^2} x_1^2 + \frac{\Delta}{\pi} x_1 - \epsilon)^2$ $+ 1 \cdot \left(1 - \frac{1}{\lambda\pi}\right) \cos(x_1) + 1 \cdot$
B	Bohachevsky	$B_2(x) = x_1^2 + 2x_2^2 - 0.3 \cos(\pi x_1) - 0.4 \cos(\pi x_2)$ $+ 0.7$
BL	Beale	$BL(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.2\Delta - x_1 + x_1 x_2^2)^2$ $+ (2.62\Delta - x_1 + x_1 x_2^2)^2$
BO	Booth	$BO(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - \Delta)^2$
CV	Colville	$CV(x) = 1.0 \cdot (x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_2 - 1)^2$ $+ 9 \cdot (x_2^2 - x_3)^2 + 1.0 \cdot 1 \cdot ((x_2 - 1)^2 + (x_3 - 1)^2)$ $+ 19.8(x_2 - 1)(x_3 - 1)$
DJ	De Jong	$DJ(x) = x_1^2 + x_2^2 + x_3^2$
DP	Dixon & Price	$DP_n(x) = (x_1 - 1)^2 + \sum_{i=1}^n i(x_i^2 - x_{i-1})^2$
GP	Goldstein & Price	$GP(x) = \left(1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 13x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right)$ $\cdot (30 + (2x_1 - 2x_2)^2 (18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1 x_2 + 27x_2^2))$

---

G	Griewank	$G_n(x) = \sum_{i=1}^n \frac{x_i}{\sqrt{i}} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
HM	Hump	$HM(x) = \frac{1}{4}x_1^2 - 2.1x_1^4 + \frac{1}{5}x_1^6 + x_1x_2 - \frac{1}{4}x_2^2 + \frac{1}{5}x_2^4$
L	Levy	$L_n(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} \left[ (y_i - 1)^2 \left( 1 + \sin^2(\pi y_i + 1) \right) \right] + (y_n - 1)^2 \left( 1 + \sin^2(\pi y_n) \right), y_i = 1 + \frac{x_i - 1}{4}$
MT	Matyas	$MT(x) = 100(x_1^2 + x_2^2) - 0.45x_1x_2$
RT	Rastrigin	$RT_n(x) = 1 \cdot n + \sum_{i=1}^n (x_i^2 - 1 \cdot \cos(\pi x_i))$
R	Rosenbrock	$R_n(x) = \sum_{i=1}^{n-1} [100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$
SC	Schwefel	$SC_n(x) = 418.9829n - \sum_{i=1}^n (x_i \sin \sqrt{ x_i })$
SR	Sphere	$SR_n(x) = \sum_{i=1}^n x_i^2$
SS	Sum Square	$SS_n(x) = \sum_{i=1}^n ix_i^2$
T	Trid	$T_n(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$
Z	Zakharov	$Z_n(x) = \sum_{i=1}^n x_i^2 + \left( \sum_{i=1}^n \cdot \Delta i x_i \right)^2 + \left( \sum_{i=1}^n \cdot \Delta x_i \right)^4$

---

## مراجع

- [1] Zainuddin, Z., Pauline, O., Function Approximation Using Artificial Neural Networks, WSEAS transaction on mathematics, June 2008, Issue 6, Vol 7.
- [2] Bromhead, D. S., Lowe, D., Multivariate functional interpolation and adaptive networks, 1998, Complex Systems, Vol. 2, pp. 321-355.
- [3] Kennedy, J.; Eberhart, R., Particle Swarm Optimization, 1995, Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942-1948. doi: 10.1109/ICNN.1995.488968
- [4] Smyth, G.K., Numerical analysis. Encyclopedia of Biostatistics, 2004. 6: p. -3762 .3767
- [5] Haan, W.J.d., Function approximation. 2011
- [6] Crevier, D., The Tumultuous History of the Search for Artificial Intelligence 1993, New York: HarperCollins Publishers.
- [7] Lisboa, P.J.G., Vellido, A., Edisbury, B., Business Applications Of Neural Networks. Vol.2000 .13 .
- [8] Poggio. T., and Girosi, F., Networks for approximation and learning, 1990, Proc. IEEE 78(9), 1484-1487 .
- [9] Rosenblatt, F., The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. Psychological Review, .1958

- [10] Yee, P. V. and Haykin, S, Regularized Radial Basis Function Networks: Theory and Applications, 2001, John Wiley. ISBN 0-471-35349-3.
- [11] Vakil-Baghmishe, M. T., Pavesic, N., Training RBF networks with selective backpropagation, 2004, Neurocomputing 62, Elsevier, pp 39-64.
- [12] Fernandez-Redondo, M. , Training Radial Basis Functions by Gradient Descent, 2006, International Joint Conference on Neural Networks, july 2006. IJCNN '06.
- [13] Li, S. T., Chen, S. C, Function Approximation using Robust Wavelet Neural Network, Proceedings. 14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002).
- [14] Sedighzade, D., Masehian, Ellips., Particle Swarm Optimization Methods, Taxonomy and Applications International Journal of Computer Theory and Engineering, 2009. 1(Δ): p. 502-486
- [15] Kennedy, J.a.E., R.C. "Particle Swarm Optimization. in Proc .IEEE Int. Conf. on N.N. .1995
- [16] Shi, Y.E., R.C. A modified particle swarm optimizer. in Proceedings of IEEE International Conference on Evolutionary Computation. .1998
- [17] Clerc, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. in Proc. congress on Evolutionary Computation. 1999. Washington, DC.
- [18] Eberhart, R.C., and Shi, Y., Computational Intelligence: Concepts to Implementations 2007: Morgan Huffman.



- [19] Eberhart, R. C., Hu, X., Human tremor analysis using particle swarm optimization.,  
proc Congress on Evolutionary Computation 1999, Washington, DC, pp 1927-  
1930. Piscataway, NJ: IEEE service Center.
- [20] Tandon, V., Closing the gap between CAD/CAM and optimization CNC and  
miling., Master's thesis, 2000, Purude School of Engineering and Technology,  
Indiana University Purude University Indianapolis.
- [21] Yoshida, H., 1999, A particle swarm optimization for reactive power and voltage  
control considering voltage stability. In G. L. Torres and A. P. Alves da Silva, Eds.,  
Proc Intl. Conf. on Intelligent System Application to Power Systems, Rio de Janiro,  
Brazil, 117-121.
- [22] Eberhart, R. C., Simpson, P. K., and Dobbins, R. W., 1996. Computational  
Intelligence PC tools. Boston, MA: Academic Press Professional.
- [23] Eberhart, R.C., and Shi, Y., Computational Intelligence: Concepts to  
Implementations 2004: Morgam Huffman.
- [24] Zhang, Y.-N., Hu, Q.-N. and Teng, H.-F., Active target particle swarm  
optimization: Research Articles. J. of Concurrency and Computation: Practice &  
Experience, 2008. 20(1): p. 40-29
- [25] Xie, X.-F., Zhang, W.-J. and Yang, Z.-L., Adaptive Particle Swarm Optimization on  
Individual Level, in Int. Conf .On Signal Processing (ICSP) 2002. p. 1218-1218
- [26] Pant, M., Thangaraj, R. and Abraham, A., Particle Swarm Optimization Using  
Adaptive Mutation. IEEE/DEXA'08, 2008: p. 523-519

- [27] Alviar, J.-B., Peña, J., and Hincapié, R., Subpopulation best rotation: a modification on PSO. *Revista Facultad de Ingeniería*, 2007. 40: p. 122-118
- [28] Kennedy, J., and Eberhart, R.C., A discrete binary version of the particle swarm algorithm, in *Int. IEEE Conf. on Systems, Man, and Cyber.* 1997. p.4108 - 4104 .
- [29] Shi, Y., and Krohling, R. A. Co-evolutionary particle swarm optimization to solve min-max problems. in *Proc. Cong. on Evolutionary Comput.* 2002
- [30] Baskar, S.a.S., P.N. A novel concurrent particle swarm optimization. in *IEEE Cong. on Evolut.* 2004
- [31] Aguirre, A.H., Muñoz Zavala, A. E. and Diharce, E. V., and Botello Rionda, S. , COPSO: Constraints Optimization via PSO algorithm. *Comunicación Técnica, (CC/CIMAT)*, 2007
- [32] Xie, X.-F., Zhang, W.-J., and Yang, Z.-L., A Dissipative Particle Swarm Optimization, in *Cong. on Evolutionary Comput. (CEC)*2002. p. 1461-1466
- [33] Ji, C., Zhang, Y., Gao, SH., Yuan, P. and Li, Zh., Particle swarm optimization for mobile ad hoc networks clustering, in *IEEE Int. Conf. on Networking, Sensing and Control.*2004
- [34] Lian, Z., Gu, X., and Jiao, B. "A Dual Similar Particle Swarm Optimization Algorithm for Job-Shop Scheduling With Penalty. in *Proc. IEEE World Cong. on Intelligent Control and Automation.* 2006
- [35] Hu, X.a.E., R. C. "Multi objective optimization using dynamic neighborhood particle swarm optimization. in *Proc. of the IEEE/ CEC.* 2002

- [36] Kulkarni, R.V.a.V., G.K. An Estimation of Distribution Improved Particle Swarm Optimization Algorithm. in Proc. IEEE/ ISSNIP. 2007
- [37] Jun-jie, X.a.Z.-h., Xin. "An extended particle swarm optimizer. in Proc. IEEE Symp., Parallel and Distributed. 2008
- [38] Secrest, B.R.a.L., G.B. Visualizing particle swarm optimization – Gaussian particle swarm optimization. in Proc. Of Swarm Intell. Symp. (IEEE/SIS). 2003
- [39] Moraglio, A., Di Chio, C., Togelius, J. and Poli, R., Geometric Particle Swarm Optimization. J. of Artificial Evolution and Applications, 2008
- [40] Hu, X.a.E., R. C. "Multi objective optimization using dynamic neighborhood particle swarm optimization. in Proc. of the IEEE/ CEC. 2002
- [41] Pasupuleti, S.a.B., R., The Gregarious Particle Swarm Optimizer (GPSO). GECCO'06, 2006
- [42] Lam, H., T., Nikolaevna, P., N. and Quan N., T., M. The Heuristic Particle Swarm Optimization. in Proc. of annual Conf. on Genetic and evolutionary computation in Ant colony optimization, swarm Intell., and artificial immune systems,"GECCO'07". 2007
- [43] Hui, W.a.F., Q, "Improved particle swarm optimizer with behavior of distance models. J. of Computer Eng. and Applications, 2007: p. 32-30
- [44] Noel, M.M.a.J., T.C. Simulation of a new hybrid particle swarm optimization algorithm. in Proc., Of the IEEE Symp. On System Theory. 2004

- [۴۵] Wang; X.-H. and Li, J.-J. Hybrid particle swarm optimization with simulated annealing. in Proc., Of the IEEE Int. Conf. on Machine Learning and Cyber. .۲۰۰۴
- [۴۶] Lee, T.-Y., Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: a multi pass iteration particle swarm optimization approach. IEEE Trans. on Energy Conversion, ۲۰۰۷. ۲۲(۳): p. ۷۸۲-۷۷۴
- [۴۷] McNabb, A., W., Monson, C., K. and Seppi, K., D. MRPSO: MapReduce particle swarm optimization. in Proc. of the ۹th annual Conf. on Genetic and evolutionary Comput. .۲۰۰۷
- [۴۸] Yuan, L.a.Z., Z.-D., A Modified Binary Particle Swarm Optimization Algorithm for Permutation Flow Shop Problem. IEEE/ICMLC, ۲۰۰۷. ۲: p. ۹۰۷-۹۰۲
- [۴۹] Zhiming, L., Cheng, W. and Jian L. Solving constrained optimization via a modified genetic particle swarm optimization. in Proc. of Int. Conf. On Forensic applications and techniques in telecommunications, information, and multimedia and workshop. .۲۰۰۸
- [۵۰] Dou, Q., Zhou, C., Pan, G., Luo, H. and Liu, Q. , Neural particle swarm optimization for casing damage prediction. Int. Sympos. On neural networks, ۲۰۰۵. ۳۴۹۸
- [۵۱] Zhang, Q.a.M., M., A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems, in Int. IEEE Conf. on Intelligent Systems ۲۰۰۶. p. ۱۷۶- ۱۸۱

- [ΔΥ] Yang, C.a.S., D. A New Particle Swarm Optimization Technique. in Proc. Of the Int. Conf. on systems Eng., (IEEE/ISEng'05). 2005
- [ΔΥ] Brits, R., Engelbrecht, A.P., and van den Bergh, F, Niche Particle Swarm Optimization, in Department of Computer Science2005, University of Pretoria.
- [Δϕ] Li, H.-Q., and Li, L. A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems. in Proc. IEEE/IPC. 2007
- [ΔΔ] Ho, S.-Y., Lin, H.-S., Liauh, W.-H., and Ho, S.-J, "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems. IEEE Trans. on Systems, Man and Cyber, 2008. 38(2): p. 298-308
- [Δϕ] Chang, J.-F., Chu, S.-C., Roddick, J.-F. and Pan, J.-S, A parallel particle swarm optimization algorithm with communication strategies. J. of Information Science and Eng, 2005. 21(4): p. 618-629
- [ΔΥ] Koh, B.-I., Fregly, B.-J., George, A.-D. and Haftka, R.-T., Parallel asynchronous particles swarm for global biomechanical. Int J Number Methods Eng, 2005: p. 578-595
- [Δλ] Jie, J., Zeng, j. and Han, C., Self-Organization Particle Swarm Optimization Based on Information Feedback, in Advances in natural comput.: Part I-II: Second Int. conf., ICNC2006: Xi'an, China.
- [Δ9] Schoeman, I.L.a.E., A.P. A Parallel Vector-Based Particle Swarm Optimizer. in Proc. of the Int. Conf. on Neural Networks and Genetic Algorithms. 2005

- [۷۶] Yuan, Z., Jin, R., Geng, J., Fan, Y., Lao, J., Li, J., Rui, X., Fang, Z. and Sun, J. A perturbation particle swarm optimization for the synthesis of the radiation pattern of antenna array. in Proc. IEEE Conf. on Asia-Pacific. ۲۰۰۵
- [۶۰] Atyabi, A.a.P.-A., S. , Particle swarm optimization with area extension (AEPSO). IEEE/CEC, ۲۰۰۷: p. ۱۹۷۶-۱۹۷۰
- [۶۱] Özcan, E.a.Y., M. Particle Swarms for Multimodal Optimization. in Proc. of the  $\lambda$ th int. conf. on Adaptive and Natural Computing Algorithms, Part I. ۲۰۰۶
- [۶۲] Wang, X., Wang, Y., Zeng, H., and Zhou, H., Particle Swarm Optimization with Escape Velocity, in Int. Conf .on Computational Intell. and Security ۲۰۰۶. p. -۴۵۷ .۴۶۰
- [۶۳] He, S., Wu, Q.H., Wen, J.Y., Saunders, J.R. and Paton, R.C, A particle swarm optimizer with passive congregation. J. of Biosystems, ۲۰۰۴. ۷۸(۳-۱): p. ۱۴۷-۱۳۵
- [۶۴] Krink, T.V., J.S. and Riget, J .Particle swarm optimization with spatial particle extension. in Proc. of Cong. on Evolutionary Computation, (CEC'۰۲). ۲۰۰۲
- [۶۵] Higashitani, M., Ishigame, A., and Yasuda, K, Pursuit-Escape Particle Swarm Optimization. Trans. On Electrical and Electronic Eng., (IEEJ), ۲۰۰۸. ۳(۱): p. -۱۳۶ .۱۴۲
- [۶۶] Yuan, Z., Jin, R., Geng, J., Fan, Y., Lao, J., Li, J., Rui, X., Fang, Z. and Sun, J. A perturbation particle swarm optimization for the synthesis of the radiation pattern of antenna array. in Proc. IEEE Conf. on Asia-Pacific. ۲۰۰۵

- [67] Meissner, M., Schmuker, M. and Schneider, G., Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics*, 2006: p. 125-3
- [68] Ho, S.-Y., Lin, H.-S., Liauh, W.-H., and Ho, S.-J, "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems. *IEEE Trans. on Systems, Man and Cyber*, 2008. 38(2): p. 298-308
- [69] Chang, J.-F., Chu, S.-C., Roddick, J.-F. and Pan, J.-S, *A parallel particle swarm optimization algorithm with communication strategies*. *J. of Information Science and Eng*, 2005. 21(4): p. 818-809
- [70] Zhiming, L., Cheng, W. and Jian L. *Solving constrained optimization via a modified genetic particle swarm optimization*. in *Proc. of Int. Conf. On Forensic applications and techniques in telecommunications, information, and multimedia and workshop*. 2008
- [71] Seo, J.-H., Im, C.-M., Heo, C.-G., Kim, J.-K., Jung, H.-K. and Lee, C.-G., *Multimodal function optimization based on particle swarm optimization*. *IEEE Trans. on Magnetics*, 2006. 42(1): p. 98-105
- [72] Mostaghim, S., Branke, J., and Schmeck, H. *Multi-objective particle swarm optimization on computer grids*. in *in Proc. Genetic and Evolutionary Comput. Conf.*. 2007
- [73] Brits, R., Engelbrecht, A. P., and van den Bergh, F. *Solving Systems of unconstrained equations using particle swarm optimization*. in *IEEE Conf. on Sys. Man and Cyber. (SMC)*. 2002
- [74] Dou, Q., Zhou, C., Pan, G., Luo, H. and Liu, Q. , *Neural particle swarm optimization for casing damage prediction*. *Int. Sympos. On neural networks*, 2005. 3498

- [75] Zhang, Q.a.M., M., *A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multiobjective Problems*, in *Int. IEEE Conf. on Intelligent Systems* 2006. p. 176–181
- [76] Yang, C.a.S., D. *A New Particle Swarm Optimization Technique*. in *Proc. Of the Int. Conf. on systems Eng., (IEEE/ISEng'05)*. 2005
- [77] Brits, R., Engelbrecht, A.P., and van den Bergh, F, *Niche Particle Swarm Optimization*, in *Department of Computer Science* 2005, University of Pretoria.
- [78] Li, H.-Q., and Li, L. *A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems*. in *Proc. IEEE/IPC*. 2007
- [79] Glover, F., *Tabu Search - Part I*. *ORSA Journal on Computing*, 1989. 1(3): p. 190–206
- [80] Glover, F., *Tabu Search Methods in Artificial Intelligence and Operations Research*. *ORSA Artificial Intelligence*, 1987. (2)1
- [81] Li, J.Q., Pan, Q., Xie, Sh., Jia, B., Wang, Y., *A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem*. *International Journal of Computer Theory and Engineering*, 2010. 2: p. 194–199
- [82] Nakano, S., Ishigame, A., Yasuda, K., *Particle Swarm Optimization Based on the Concept of Tabu search*, in *IEEE Congress on Evolutionary Computation* 2007. p. 3263–3268



- [13] Zhang, Y., Wu, L., a hybrid TS-PSO optimization algorithm, may 2011, journal of convergence information technology, Vol 6, No 6.
- [14] Shen, Q., Shi, W., Kong, W., Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. Computational Biology and Chemistry, 2008. 32: p. 60-63
- [15] Zhang, T.Z., Y. J., Zheng, Q. P., Pardalos, P. M., A hybrid particle swarm optimization and tabu search algorithm for order planning problems of steel factories based on the Make-To-Stock and Make-To-Order management architecture. Journal of Industrial and Management Optimization (JIMO), 2011.
- Y.(1)
- [16] Bekrar, A., Chaanbane, S. Trentesaux, D., Bornschlegell, A., Pelle, J., Harmand, S., Hybrid PSO-tabu search for constrained non-linear optimization problems in International conference on swarm intelligence (ICSI 2011).2011
- [17] Ho, S. L., Yang, S., Ni, G., Wong, H. C, A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices, april 2006, IEEE transaction on magnetics, vol 42, no 4.

## Abstract

Function Approximation, is one of the topics that many science from applied mathematics to computer science, are interested in. always, there was different methods for Function Approximation and by passing the time, they became more optimized or new methods has been introduced. One of this methods that is used in Function Approximation, is artificial Neural Network. Training artificial Neural Network is truly a kind of Function Approximation, because in order to get to the proper output, the parameters and rules of neurons of different layers must be estimated. Training the artificial Neural Network, usually is done by back propagation method or gradient descent algorithm, but sometimes heuristic algorithms is used for training the Neural Networks, that could has improvements and faults either. According to the researches that has been done by scientists till now, using optimization algorithms as a tool to train the Neural Networks could save the training time and increase the convergence rate and so it result in less final error.

From the time, that optimization algorithms has been introduced, there was try to make changes and improvements and do combinations on this algorithms in order to increase the efficiency, decrease the training time and computational price, sometimes this is done by hybridizing this algorithms with other optimization methods and algorithms.

In this thesis, I proposed a hybrid method of Particle Swarm Optimization algorithm and Tabu Search algorithm, and I has used this proposed method to train the Radial basis function artificial Neural Networks in order to Function Approximation. actually by hybridizing this algorithms, I used the advantages of this algorithms, and decrease disadvantages to get in to the better and more accurate results. After that, once I used gradient descent algorithm to train the Neural Network and in another time, I used classic Particle Swarm Optimization to train the Neural Network. This is done to certificate the results of proposed algorithm and make comparison between the given results of this proposed algorithm and results of training the Neural Network by gradient descent algorithm and Particle Swarm Optimization algorithm. To do this, I've done selecting some of parameters of Particle Swarm Optimization algorithm by Tabu Search algorithm, and analysis the result of this method on training the RBF Neural Network, and compared the results by gradient descent method and classic PSO algorithm on training the RBF Neural Network for Function Approximation.

Keywords: Neural Networks, Function Approximation, Particle Swarm Optimization, Tabu Search