

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شریف

دانشکده مهندسی صنایع

پایان نامه به عنوان تحقق بخشی از شرایط دریافت درجه کارشناسی ارشد

گرایش بهینه‌سازی سیستم‌ها

**الگوریتم شبه مدل محور برای مسائل بهینه سازی شبیه سازی نیمه گران قیمت بر  
مبنای شبکه عصبی مصنوعی**

نگارش:

محمد بهبهانی

استاد راهنما:

دکتر سید تقی اخوان نیاکی

تابستان ۹۷

## تصویب نامه

به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی صنایع

پایان نامه کارشناسی ارشد

عنوان: الگوریتم شبه مدل محور برای مسائل بهینه سازی شبیه سازی نیمه گران قیمت بر

مبنای شبکه عصبی مصنوعی

نگارش: محمد بهبهانی

کمیته ممتحنین:

امضاء..... استاد راهنما: جناب آقای دکتر سید تقی اخوان نیاکی

امضاء..... استاد راهنمای همکار: جناب آقای دکتر هاشم محلوجی

استاد مدعو: جناب آقای دکتر حمیدرضا اسکندری

امضاء..... تاریخ:.....



## اظهارنامه

(اصالت متن و محتوای پایان‌نامه کارشناسی ارشد)

عنوان پایان‌نامه: .....

نام استاد راهنما: ..... نام استاد راهنمای همکار: ..... نام استاد مشاور: .....

این جانب ..... اظهار می‌دارم:

- ۱- متن و نتایج علمی ارائه شده در این پایان‌نامه اصیل بوده و منحصراً توسط این جانب و زیر نظر استادان (راهنما، همکار و مشاور) نام‌برده شده در بالا تهیه شده است.
- ۲- متن پایان‌نامه به این صورت در هیچ جای دیگری منتشر نشده است.
- ۳- متن و نتایج مندرج در این پایان‌نامه، حاصل تحقیقات این جانب به عنوان دانشجوی کارشناسی ارشد دانشگاه صنعتی شریف است.
- ۴- کلیه مطالبی که از منابع دیگر در این پایان‌نامه مورد استفاده قرار گرفته، با ذکر مرجع مشخص شده است.

نام دانشجو: .....

تاریخ

امضا

نتایج تحقیقات مندرج در این پایان‌نامه و دستاوردهای مادی و معنوی ناشی از آن (شامل فرمول‌ها، توابع کتابخانه‌ای، نرم‌افزارها، سخت‌افزارها و مواردی که قابلیت ثبت اختراع دارد) متعلق به دانشگاه صنعتی شریف است. هیچ شخصیت حقیقی یا حقوقی بدون کسب اجازه از دانشگاه صنعتی شریف حق فروش و ادعای مالکیت مادی یا معنوی بر آن یا ثبت اختراع از آن را ندارد. همچنین کلیه حقوق مربوط به چاپ، تکثیر، نسخه‌برداری، ترجمه، اقتباس و نظائر آن در محیط‌های مختلف اعم از الکترونیکی، مجازی یا فیزیکی برای دانشگاه صنعتی شریف محفوظ است. نقل مطالب با ذکر ماخذ بلامانع است.

نام دانشجو: .....

تاریخ

امضا

نام استادان راهنما: .....

تاریخ

امضا

تقدیم به

پدر و مادرم

## تقدیر و تشکر

در تهیه و تنظیم این پایان نامه از پشتیبانی افرادی چند بهرمنند بوده‌ام که ذکر نام آنها باعث افتخار اینجانب است. در این مورد جا دارد که از کمک و راهنمایی استاد ارجمندم جناب آقای دکتر نیاکی در زمینه‌های علمی و ادبی پایان نامه و همچنین از نظرات و پیشنهادهای ارزشمند جناب آقای دکتر محلوجی یاد کنم و بدینوسیله نهایت تشکر و قدردانی را داشته باشم. همچنین، از زحمات بی‌شمار پدر و مادر عزیزم نهایت قدردانی را دارم چرا که انجام این پژوهش بدون حضور گرم و دلسوزانه ایشان میسر نبود.

## چکیده:

به دنبال پیشرفت روز افزون دانش تصمیم‌گیری، به تدریج مساله‌هایی در حوزه سیستم‌های تصادفی مطرح می‌شوند که تحلیل دقیقشان علیرغم قبول فرضیات بسیار، متکی بر روابط پیچیده ریاضی است. در این مواقع، شبیه‌سازی یکی از معروف‌ترین ابزارهایی است که میتوان به منظور شناسایی رفتار یک سیستم تصادفی به کار گرفت. مدل شبیه‌سازی، علاوه بر آنکه وابسته به روابط پیچیده ریاضی نیست، بلکه بی‌نیاز از فرضیات محدود کننده‌ای است که به سیستم مورد بررسی تحمیل می‌شوند. از دو رویکرد مدل محور و شبه مدل محور برای حل مسائل بهینه‌سازی شبیه‌سازی استفاده می‌شود. رویکرد مدل محور هنگامی که زمان شبیه‌سازی کوتاه است به کار می‌رود در صورتی که با افزایش زمان شبیه‌سازی، ارزیابی نقاط بدست آمده از رویکرد مدل محور هزینه بر شده و به روش‌های شبه مدل محور روی می‌آوریم که در آنها با استفاده از تعداد محدودی نقطه شبه مدل اولیه بر روی فضای جواب برازش شده و پس از کسب اطمینان از صحت شبه مدل، فرایند بهینه‌سازی آغاز می‌شود. در این میان مسائلی با زمان شبیه‌سازی متوسط وجود دارد (نیمه‌گران قیمت) که رویکرد مدل محور با توجه به نیاز اجرای شبیه‌سازی‌های مکرر برای بهینه‌سازی آنها به صرفه نیست. از طرفی با بکارگیری روش‌های شبه مدل محور در این گونه مسائل، هزینه‌های اعتبار سنجی شبه مدل بر هزینه‌های شبیه‌سازی غلبه کرده و تمایل را برای استفاده از این رویکرد کاهش می‌دهد. به عبارت دیگر شبیه‌سازی به حدی زمان بر نیست که رویکرد شبه مدل محور برای آنها به صرفه تر باشد. برای حل این مشکل به تازگی روشی تحت عنوان الگوریتم دو مرحله‌ای معرفی شده است که با ترکیب دو رویکرد مدل محور و شبه مدل محور در دو مرحله و استفاده از شبه مدل کراگینگ به حل چنین مسائلی در زمان بسیار مناسب می‌پردازد. هدف ما از این پژوهش ضمن پیاده‌سازی شبه مدل شبکه عصبی مصنوعی به عنوان شبه مدل الگوریتم، بررسی بیشتر دامنه ویژگی‌های مسائلی است که این الگوریتم می‌تواند برای آنها حداکثر کارایی را همراه داشته باشد..

**کلمات کلیدی:** شبیه‌سازی نیمه‌گران قیمت، بهینه‌سازی شبیه‌سازی، الگوریتم شبه مدل محور، شبکه عصبی مصنوعی

## فهرست مطالب

۲	۱. فصل اول: مقدمه .....
۲	۱-۱. تاریخچه بهینه‌سازی شبیه‌سازی .....
۲	۲-۱. شبیه سازی .....
۳	۱-۲-۱. مدلسازی سیستم .....
۴	۲-۲-۱. موارد کاربرد شبیه سازی .....
۵	۳-۲-۱. مزایا و معایب شبیه‌سازی .....
۶	۴-۲-۱. گام‌های شبیه سازی .....
۱۰	۲. فصل دوم: بهینه‌سازی شبیه‌سازی .....
۱۳	۱-۲. دسته‌بندی ساختاری مدل و روش‌های بهینه‌سازی شبیه‌سازی .....
۱۵	۲-۲. روشهای مدل محور .....
۱۶	۱-۲-۲. روشهای گرادیان محور .....
۱۶	۲-۲-۲. روشهای بهینه سازی تصادفی .....
۱۶	۳-۲-۲. روشهای آماری .....
۱۷	۴-۲-۲. روشهای فرا ابتکاری .....
۱۹	۵-۲-۲. روش بهینه سازی مسیر نمونه .....
۲۰	۳-۲. روشهای شبه مدل محور .....
۲۱	۱-۳-۲. شبه مدل رگرسیونی .....



- ۲۲..... ۲-۳-۲. شبه مدل شبکه عصبی مصنوعی
- ۲۸..... ۳-۳-۲. شبه مدل کراگینگ
- ۳۰..... ۴-۲. طراحی آزمایش
- ۳۰..... ۱-۴-۲. طرح عاملی کامل
- ۳۱..... ۲-۴-۲. طرح عاملی دو سطحی
- ۳۱..... ۳-۴-۲. طرح فوق مکعب لاتین
- ۳۲..... ۵-۲. اعتبارسنجی شبه‌مدل
- ۳۳..... ۱-۵-۲. روش ضریب تعیین
- ۳۴..... ۲-۵-۲. روش اعتبار سنجی متقاطع
- ۳۴..... ۳-۵-۲. روش بوتسترپ
- ۳۶..... ۲. فصل سوم: مرور ادبیات
- ۳۶..... ۱-۳. مرور بر ادبیات بهینه‌سازی شبیه‌سازی
- ۳۸..... ۲-۳. مرور بر ادبیات بهینه‌سازی شبیه‌سازی گران قیمت
- ۴۰..... ۳-۳. مرور ادبیات مقایسه شبه مدل کراگینگ و شبکه عصبی مصنوعی
- ۴۴..... ۴. الگوریتم بهینه‌سازی شبه‌مدل محور پیشنهادی
- ۴۵..... ۱-۴. تعریف متغیرهای مسئله (فاز اول - گام ۱)
- ۴۵..... ۲-۴. توسعه طرح آزمایش اولیه (فاز اول - گام ۲)
- ۴۷..... ۳-۴. اجرای مدل شبیه‌سازی برای نقاط نمونه اولیه (فاز اول - گام ۳)

- ۴-۴. برآزش شبه‌مدل شبکه عصبی (فاز اول - گام ۴) ..... ۴۷
- ۵-۴. اعتبار سنجی شبه‌مدل (فاز اول - گام ۵) ..... ۴۸
- ۶-۴. اجرای مدل بهینه‌سازی برای یافتن نقاط جدید (فاز اول - گام ۶) ..... ۴۸
- ۷-۴. تعویض مدل شبیه‌سازی با شبه‌مدل (فاز دوم) ..... ۴۸
- ۸-۴. تفاوت ویژگی‌های الگوریتم بهینه‌سازی تجمع ذرات مورد استفاده در فاز اول و دوم ..... ۴۹
۵. بررسی عددی ..... ۵۱
- ۱-۵. اجرای الگوریتم برای تابع اسفیر ..... ۵۲
- ۲-۵. اجرای الگوریتم برای تابع گریوانک ..... ۵۳
- ۳-۵. اجرای الگوریتم برای تابع اسچافر اف ۶ ..... ۵۵
- ۴-۵. اجرای الگوریتم برای تابع راسترچین ..... ۵۷
- ۵-۵. اجرای الگوریتم برای تابع رسنبراک ..... ۵۸
- ۶-۵. مقایسه بکارگیری شبه‌مدل شبکه عصبی مصنوعی در برابر شبه‌مدل کراگینگ ..... ۶۰
۶. نتیجه‌گیری و تعیین پژوهش‌های آتی ..... ۶۳
۷. منابع ..... ۶۵

## فهرست جداول

- جدول ۵-۱ راه اندازی مقدار اولیه پارامترها ..... ۵۱
- جدول ۵-۲ نتایج بدست آمده از اجرای الگوریتم برای تابع اسفیر ..... ۵۳
- جدول ۵-۳ نتایج بدست آمده از اجرای الگوریتم برای تابع گریوانک ..... ۵۴
- جدول ۵-۴ نتایج بدست آمده از اجرای الگوریتم برای تابع اسچافر ..... ۵۶
- جدول ۵-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع راسترچین ..... ۵۸
- جدول ۵-۶ نتایج بدست آمده از اجرای الگوریتم برای تابع رسنبراک ..... ۵۹
- جدول ۵-۷ نتایج عددی ..... ۶۰
- جدول ۵-۸ تحلیل واریانس ..... ۶۱

## فهرست شکل‌ها

- شکل ۱-۱ گام‌های اساسی شبیه‌سازی ..... ۷
- شکل ۱-۲ مدل کلی شبیه‌سازی ..... ۱۱
- شکل ۲-۲ فرآیند بهینه‌سازی شبیه‌سازی ..... ۱۱
- شکل ۳-۲ روش‌های بهینه‌سازی شبیه‌سازی ..... ۱۴
- شکل ۴-۲ تقسیم بندی مساله های بهینه سازی شبیه سازی از لحاظ ساختار ..... ۱۵
- شکل ۵-۲ بهینه‌سازی شبیه‌سازی شبه مدل محور ..... ۲۱
- شکل ۶-۲ شبکه عصبی بیولوژیکی ..... ۲۳
- شکل ۷-۲ شبکه عصبی مصنوعی ساده تک لایه ..... ۲۴
- شکل ۸-۲ شبکه عصبی مصنوعی دو لایه ..... ۲۵
- شکل ۹-۲ نمونه ای از توزیع نقاط آزمایش در یک طرح آزمایش فوق مکعب لاتین ..... ۳۲
- شکل ۱-۳ سطح پاسخ تابع  $f_1(x)$  ..... ۴۰
- شکل ۲-۳ سطح پاسخ تابع  $f_2(x)$  ..... ۴۰
- شکل ۳-۳ نمایش روش اعتبار سنجی متقاطع کنارگذاری تکی ..... ۴۲
- شکل ۱-۴ الگوریتم بهینه‌سازی شبیه‌سازیهای نیمه گران قیمت ..... ۴۶

# فصل اول:

## مقدمه پژوهش

## ۱. مقدمه

### ۱-۱. تاریخچه بهینه‌سازی شبیه‌سازی

ابتدای بهینه‌سازی شبیه‌سازی مربوط به استفاده از شبیه‌سازی برای یافتن پارامترهای ورودی بهتری بوده که منجر به خروجی مطلوب‌تری شود. به همین دلیل با در نظر گرفتن اینکه در ابتدا واژه "بهینه‌سازی شبیه‌سازی"<sup>۱</sup> به صورت متداول توسط محققان استفاده نشده است، ریشه پیدایش بهینه‌سازی شبیه‌سازی در ابتدای تاریخچه شبیه‌سازی قابل جستجو است. بهینه‌سازی شبیه‌سازی که هدف نهایی اکثر کاربران شبیه‌سازیست، تاریخچه غنی و بسیار نزدیک به ۵۰ سال برگزاری کنفرانس زمستانه شبیه‌سازی<sup>۲</sup> را داراست. با این وجود تاریخچه یافتن جواب بهتر با استفاده از شبیه‌سازی به بیش از ۵۰ سال باز می‌گردد. بر همین اساس در ابتدای پژوهش به بیان اصول شبیه‌سازی شامل مدلسازی سیستم، موارد کاربرد شبیه‌سازی، مزایا و معایب شبیه‌سازی و بررسی گام‌های اساسی شبیه‌سازی پرداخته خواهد شد و پس از آن در فصل ۲ به بررسی اجمالی بهینه‌سازی شبیه‌سازی و تکنیک‌های آن می‌پردازیم. همچنین در این بخش روش‌های مدل محور و شبه‌مدل محور و شبه‌مدل‌ها به ویژه شبه‌مدل شبکه عصبی مصنوعی که کاربرد مهمی در این پروژه دارد معرفی می‌شود. در فصل ۳ پس از مرور اجمالی بر ادبیات بهینه‌سازی شبیه‌سازی، در مورد پژوهش‌های صورت گرفته در حوزه بهینه‌سازی شبیه‌سازی گران قیمت و سپس به مقایسات موجود میان شبه‌مدل کراگینگ و شبکه عصبی اشاره خواهیم پرداخت. فصل چهارم به توضیح توسعه مدل اختصاص دارد و در نهایت به نتایج بدست آمده و مطالعات آتی می‌پردازیم.

### ۱-۲. شبیه‌سازی

یکی از بهترین روش‌های تحقیق در عملیات برای تجزیه و تحلیل سیستم‌های بزرگ شبیه‌سازیست. شبیه‌سازی یک ابزار مدلسازی است که به دنبال تعیین رابطه بین متغیرهای ورودی و متغیرهای پاسخ

---

<sup>۱</sup> Simulation optimization

<sup>۲</sup> Winter Simulation Conference

سیستم است. شبیه‌سازی چه به صورت دستی چه به صورت کامپیوتری، تقلیدی از عملکرد سیستم واقعی با گذشت زمان است که به ایجاد ساختگی تاریخچه سیستم و بررسی آن به منظور دستیابی به نتیجه‌گیری در مورد ویژگی‌های عملکرد واقعی آن می‌پردازد. شبیه‌سازی اصولاً به شکل مجموعه‌ای از فرض‌های مربوط به عملکرد سیستم در چارچوب رابطه‌های ریاضی و منطقیست. شبیه‌سازی برای بررسی سیستم‌های در دست طراحی نیز استفاده می‌شود. در نتیجه شبیه‌سازی، هم به منزله ابزار تحلیل، برای پیش‌بینی تاثیر تغییرات سیستم‌های موجود و هم به عنوان ابزار طراحی، به منظور پیش‌بینی عملکرد سیستم جدید کاربرد دارد [۱].

### ۱-۲-۱. مدل‌سازی سیستم

مجموعه‌ای یکپارچه از اشیاء که در راستای تحقق هدفی معین فعالیت می‌کنند را سیستم می‌گویند. هر سیستم، اغلب تحت تاثیر تغییراتی است که در خارج از آن اتفاق می‌افتد. در این حالت بیان می‌شود که این تغییرات در پیرامون سیستم اتفاق افتاده‌اند. سیستم‌ها به دو دسته گسسته و پیوسته تقسیم می‌شوند. اگر در سیستمی متغیرهای حالت آن در نقاط گسسته‌ای از زمان تغییر کنند آن را سیستم گسسته گوئیم و اگر متغیرهای حالت آن به صورت مداوم تغییر کنند، سیستم پیوسته محسوب می‌شود [۲].

مدل شبیه‌سازی نمایشی از یک سیستم واقعی است. برای سنجش میزان مناسب بودن مدل به وسیله مقایسه رفتار مدل با رفتار سیستم قابل سنجش است و این در حالی است که مدل مورد نظر نباید بیش از حد پیچیده شود. به این ترتیب یک مدل توانایی ارائه تمام جزئیات سیستم را ندارد و تنها می‌توان آن را به عنوان جانشینی از سیستم استفاده کرد. مدل‌های شبیه‌سازی را می‌توان به مدل‌های ایستا یا پویا، قطعی یا تصادفی، گسسته یا پیوسته تقسیم بندی کرد. مدل ایستای شبیه‌سازی که به شبیه‌سازی مونت کارلو<sup>۱</sup> معروف است، بیان‌کننده سیستم در لحظه‌ای خاص از زمان است. مدل‌های پویای شبیه‌سازی، گذشت زمان در سیستم را در نظر می‌گیرند. مدل‌های قطعی شامل مجموعه مشخصی از ورودی‌ها هستند که به مجموعه یکتایی از خروجی‌ها می‌انجامند. از طرف دیگر، مدل‌های تصادفی شامل یک یا چند متغیر تصادفی، به عنوان متغیرهای ورودی هستند.

---

<sup>۱</sup> Mont Carlo

یک مدل برای بررسی یک سیستم تعریف می‌شود که شاید تمام جزئیات آن را شامل نشود ولی با در نظر گرفتن جزئیات اساسی آن به عنوان جانشین مناسبی برای تحلیل و استخراج نتایج معتبر از عملکرد سیستم به کار می‌رود. مدل شبیه سازی نیز به عنوان یک مدل ریاضی از سیستم همین هدف را دنبال می‌کند. سیستم‌های بسیاری در دنیای واقعی وجود دارند که برای بررسی رفتار آنها می‌توان از ابزار شبیه سازی به جای صرف هزینه و زمان در بررسی سیستم اصلی بهره جست. به عنوان مثال تعیین راندمان و کارایی نهاده‌های مختلف یک کارگاه تولید، تصمیم در مورد تعداد باجه‌های یک بانک برای کمینه‌سازی زمان انتظار مراجعین، یافتن بهترین سطح از موجود کالا در زنجیره عرضه یک شرکت خدماتی و کاربردهای بسیار دیگری از این دست با مدلسازی شبیه‌سازی قابل بررسی هستند.

#### ۲-۲-۱. موارد کاربرد شبیه سازی

به طور کلی پژوهشگران هدف از انجام شبیه‌سازی را بسیار وسیع می‌دانند. اصلی‌ترین موارد به قرار زیرند:

۱. استفاده برای شناخت رفتار سیستم در مقابله با برنامه‌ریزی‌های جدید پیش از اجرای آن‌ها و اصلاح برنامه‌ها در صورت نیاز
۲. شناخت اصلی‌ترین و مهم‌ترین متغیرهای سیستم و روابط حاکم بر آن‌ها با ایجاد تغییر در متغیرهای ورودی و تحلیل حساسیت بر روی آن‌ها
۳. تشخیص روابط بین ورودی‌ها و خروجی‌های سیستم برای یافتن بهترین ترکیب ورودی برای نیل به اهداف مورد انتظار از سیستم
۴. تعیین روابط در سیستم‌های پیچیده و چگونگی اثرگذاری نهاده‌های سیستم بر یکدیگر به کمک شبیه‌سازی
۵. استفاده از خروجی‌های شبیه‌سازی برای پیش‌بینی وضعیت سیستم در آینده و برنامه‌ریزی برای اصلاحات یا بهبود عملکرد آن

در مواردی اندک، این امکان وجود دارد که سیستم‌های ساده را از طریق مدل‌های ریاضی، همانند معادلات دیفرانسیل و یا روش‌های جبری تحلیل کرد. اما از طرف دیگر، سیستم‌های پیچیده زیادی نیز وجود دارند که



حل ریاضی مدل‌هایشان غیر ممکن است. در این گونه موارد، شبیه سازی کامپیوتری، یکی از راه حل‌های ممکن برای تقلید عملکرد سیستم واقعی است. در حقیقت، در دسترس بودن زبان‌های ویژه شبیه سازی و توانایی‌های محاسباتی گسترده، این مبحث را به صورت یکی از رایج ترین و پذیرفته ترین ابزار تحقیق در عملیات و تحلیل سیستمها در آورده است.

### ۱-۲-۳. مزایا و معایب شبیه سازی

با وجود اینکه شبیه سازی، ابزار مناسبی برای تحلیل سیستم‌های پیچیده است، اما مزایا و معایب آن قبل از بکارگیری باید در نظر گرفته شود. مزایای مطرح شده توسط کارسن و بنکس<sup>۱</sup> عبارتند از:

- برای طرح‌ها و برنامه‌های پیشنهادی می‌توان مدل شبیه‌سازی را بارها مورد استفاده قرار داد.
- در تحلیل هر سیستم پیشنهادی ابزار شبیه‌سازی می‌تواند قدرتمند باشد.
- در مقایسه با تحلیل سیستم واقعی، مدل شبیه‌سازی بسیار کم‌هزینه‌تر است.
- روش شبیه‌سازی در مقابله با روش‌های تحلیلی بسیار آسان‌تر هستند.
- برخلاف روش‌های تحلیلی که اغلب به فرض‌های ساده‌کننده نیاز دارند، مدل شبیه‌سازی بدون نیاز به این فرض‌ها هر معیار سنجش عملکرد سیستم را محاسبه می‌کنند.
- در برخی مسائل شبیه‌سازی تنها راه تجزیه و تحلیل سیستم است.

و همچنین در ادامه معایب شبیه سازی از نظر اشمید و تیلور<sup>۲</sup> بیان شده است:

- طراحی و مدل‌سازی شبیه‌سازی سیستم‌های پیچیده و اعتبارسنجی آن‌ها هزینه‌بر و زمان‌بر است.
- هزینه‌ی اجرای آن برای مدل‌هایی که نیاز به تعداد اجرای بالا دارند بالاست.
- تکنیک شبیه‌سازی ممکن است در برخی مسائل به جای روش‌های تحلیل به کار رود در حالی که نیاز به استفاده از این ابزار نیست.

---

<sup>۱</sup> Carson and banks

<sup>۲</sup> Eshmid and Tylor

گام‌های اصلی پیاده‌سازی یک مدل شبیه‌سازی را طبق نظر بنکس و کارسن<sup>۱</sup> [۳] می‌توان طبق شکل ۱-۱ توضیح داد. این گام‌ها با کمی تفاوت توسط دیگر محققین به همین شکل بیان شده است.

گام ۱. صورت‌بندی مسئله: در ابتدا لازم است مسئله توسط سیاست‌گذار یا تحلیل‌گر به روشنی مشخص شود و درک متقابل صحیح شکل گیرد. در برخی موارد نیاز می‌شود که صورت‌های دیگری از مسئله بعد از بررسی‌های اولیه ارائه شود.

گام ۲. تعیین اهداف و طرح کلی: با توجه به صورت‌بندی مسئله باید اهداف شبیه‌سازی تعیین شود که از اجرای مدل شبیه‌سازی به دنبال دستیابی به چه اطلاعاتی از مدل هستیم. در این مرحله باید دلایل استفاده از شبیه‌سازی برای تحلیل مشخص شود. طرح کلی اجرا هزینه، زمان و افراد مورد نیاز برای انجام پروژه شبیه‌سازی را تعیین می‌کند.

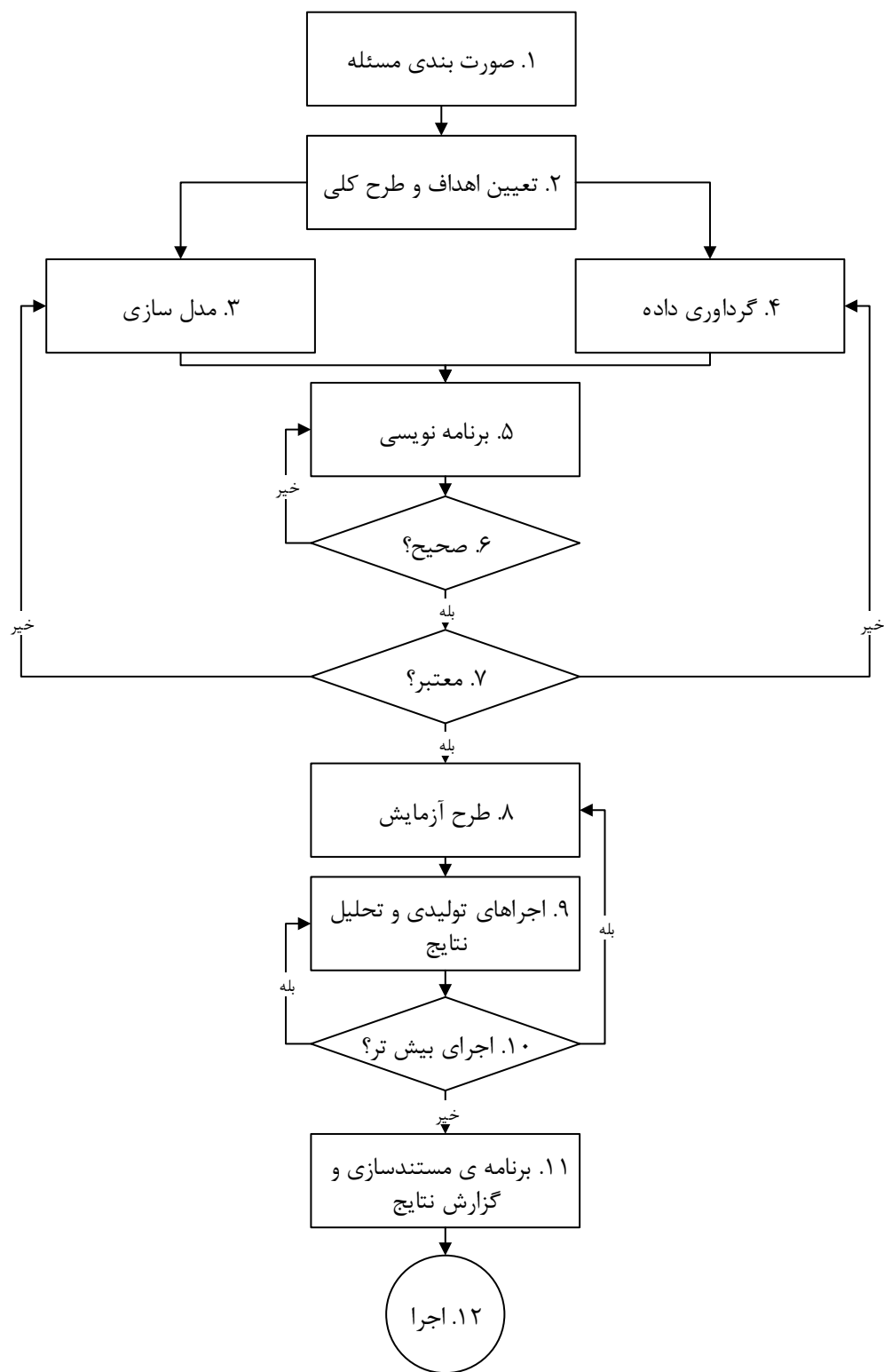
گام ۳. مدل‌سازی: در این گام با تشخیص فرض‌های اساسی تعریف‌کننده‌ی سیستم و شناخت پیشامدها و حالات آن به مدل‌سازی سیستم پرداخته می‌شود. در نظر گرفتن خصوصیات اساسی سیستم (نه تمام جزئیات آن) در کنار اهداف مورد نظر از اجرای شبیه‌سازی می‌تواند ایجاد مدلی موفق از سیستم واقعی بیانجامد.

گام ۴. گردآوری داده: این مرحله همزمان با مدل‌سازی انجام می‌شود و ضمن غنی‌سازی مدل لازم است که داده‌های متناسب با آن جمع‌آوری شود. در این مرحله اهداف شبیه‌سازی که در گام ۲ معرفی شدند می‌تواند نوع داده‌های مورد نیاز را به خوبی تعیین کند.

گام ۵. برنامه‌نویسی: به کمک مدل‌سازی و داده‌های گردآوری شده لازم است با یکی از زبان‌های برنامه‌نویسی یا نرم‌افزارهای موجود در حوزه‌ی شبیه‌سازی به پیاده‌سازی مدل در کامپیوتر پرداخت.

---

<sup>۱</sup> Carson and banks



شکل ۱-۱ گام‌های اساسی شبیه‌سازی [۳]

گام ۶. تصحیح برنامه: در این مرحله لازم است با اجرای شبیه‌سازی این موضوع روشن شود که آیا برنامه به درستی اجرا می‌شود یا خیر؟ چنانچه در اجرای برنامه خللی وجود داشته باشد لازم است، برنامه‌نویسی باید مورد بازبینی و اصلاح قرار گیرد.

گام ۷. اعتبارسنجی مدل: بعد از اجرای شبیه‌سازی در کامپیوتر لازم است بررسی شود آیا مدل شبیه‌سازی تقلیدی مناسب از عملکرد سیستم واقعی دارد یا خیر؟ تکنیک‌هایی مانند تغییر پارامترهای ورودی در محدوده‌ی قابل قبول و بررسی خروجی و یا ثابت کردن چند متغیر و بررسی نتایج حاصل‌شده در این گام مورد استفاده قرار می‌گیرد.

گام ۸. طرح آزمایش: در این مرحله لازم است مواردی که برای اجرای شبیه‌سازی مدنظر هستند، تعیین شوند. هم‌چنین طول دوره‌ی راه‌اندازی، دوره‌ی شبیه‌سازی و تعداد اجراها در این گام باید مشخص کرد.

گام ۹. اجراهای تولیدی و تحلیل نتایج: مدل شبیه‌سازی چندین مرتبه مورد تکرار واقع می‌شود و از این نتایج این تکرارها می‌توان برآورد خوبی از معیارهای عملکرد به‌دست آورد.

گام ۱۰. لزوم ایجاد اجراهای بیش‌تر: تحلیل‌گر ممکن است با طرح‌های جدید آزمایش اجراهای بیش‌تری از مدل شبیه‌سازی را راه‌اندازی کند.

گام ۱۱. برنامه‌ی مستندسازی و گزارش نتایج: پس از اجراهای مورد نظر مدل شبیه‌سازی و استخراج نتایج مورد نظر لازم است تمامی مراحل قبل به‌طور شفاف مستندسازی و گزارش شوند. با انجام درست این کار می‌توان از بسیاری چالش‌های ممکن در آینده برای اجراهای مجدد شبیه‌سازی جلوگیری کرد. هم‌چنین برای انتخاب ترکیب متغیرهای ورودی که معیار عملکرد مورد نظر را بهینه می‌کنند مستندسازی لازم است.

گام ۱۲. اجرا: چنانچه گام‌های قبل به درستی پیاده‌سازی شده باشند و سیاست‌گذار طرح، درک مناسبی از مدل و خروجی‌های آن داشته باشد، اجرای موفق‌تری از طرح حاصل خواهد شد.

## **فصل دوم:**

# **بهینه‌سازی شبیه‌سازی**

## ۲. بهینه‌سازی شبیه‌سازی

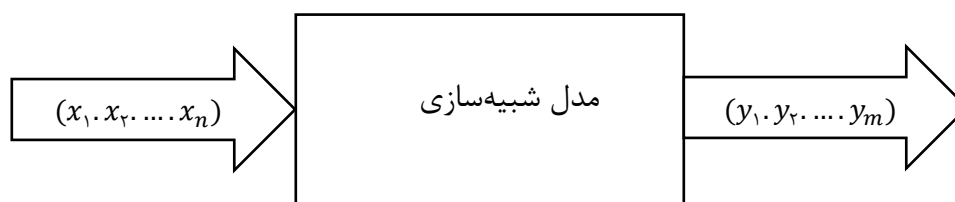
طبق مطالب گفته شده در فصل ۱، حل تحلیلی سیستم‌های تصادفی بزرگ، به دلیل رابطه پیچیده حاکم بین اجزای سیستم، بسیار مشکل است و حتی در مواردی امکانپذیر نیست. یکی از مشخصه‌های اصلی در بسیاری از سیستم‌های تصادفی دنیای واقعی، وجود مولفه‌های تصادفی غیر قابل کنترل (ناخواسته) است که باعث پیچیده تر شدن تحلیل این سیستم‌ها شده است. در این مواقع، مدل شبیه سازی، به عنوان یکی از کارآمدترین ابزارهای تجزیه و تحلیل سیستم شناخته می‌شود. این ابزار، علاوه بر داشتن هزینه محاسباتی پایین‌تر نسبت به سایر روشها، در اغلب موارد باعث بهبود در تخمین ارزیابی عملکرد سیستم می‌شود.

در یک مطالعه شبیه‌سازی، تحلیل‌گر ممکن است علاوه بر بررسی عملکرد سیستم فعلی، مایل به ارزیابی عملکرد آن به ازای مقادیر متنوعی از پارامترهای ورودی سیستم باشد. در این صورت، تحلیل‌گر به انجام آزمایشهای شبیه سازی اقدام خواهد کرد. منظور از یک آزمایش شبیه سازی، اجرای مدل شبیه سازی به دفعات، با هدف بررسی حساسیت متغیر پاسخ شبیه سازی (خروجی شبیه‌سازی) نسبت به تغییرات محسوس در متغیرهای ورودی شبیه‌سازی (پارامترهای شبیه سازی) است. در همین راستا، تحلیل‌گر ممکن است مایل به تعیین ترکیبی از متغیرهای ورودی سیستم باشد که به ازای آن، بهترین عملکرد سیستم حاصل میشود (به عبارت بهتر، تحلیل‌گر به دنبال بیشینه یا کمینه کردن متغیر پاسخ سیستم است). به این نوع مساله بهینه سازی که در حیطه سیستم‌های قطعی پیچیده و یا سیستم‌های تصادفی مطرح می‌شود، مساله بهینه‌سازی شبیه‌سازی گفته می‌شود. بهینه‌سازی شبیه‌سازی موقعیتی برای تصمیم‌گیرنده ایجاد می‌کند که به ازای حالت‌های متفاوت از متغیرهای ورودی بدون بهره‌گیری از تمام امکانات موجود، به بررسی خروجی‌ها و اهداف مورد نظر خود بپردازد و با تحلیل حساسیت بر روی آن‌ها، ترکیب مورد نظر برای بهترین عملکرد سیستم را ارائه دهد.

مثال: در یک چهارراه اصلی شهری فرض کنید در حال حاضر یک گلوگاه در ساعات شلوغ این چهارراه وجود دارد. تصمیم‌گیری مناسب برای نصب و راه‌اندازی چراغ‌های راهنمایی در این چهارراه لازم می‌باشد. مسئله تعیین زمان مناسب برای چراغ سبز برای هر طرف از چهارراه می‌باشد به صورتی که زمان انتظار کل برای خودروها در این چهارراه کمینه شود. جهت تصمیم‌گیری مناسب لازم است با اطلاع از نرخ ورود خودرو در هر

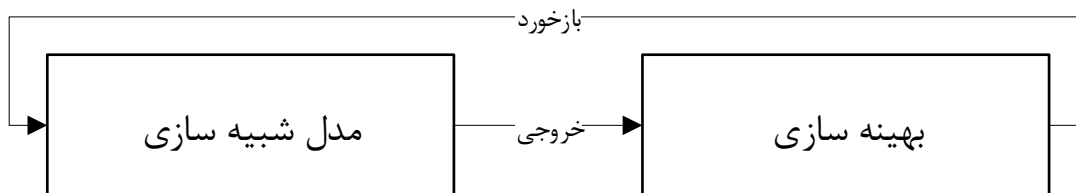
جهت این چهارراه، مدل شبیه‌سازی آن به ازای متغیرهای ورودی برای زمان چراغ سبز طراحی و اجرا شود. با بدست آوردن زمان انتظار خودروها در هر جهت به عنوان خروجی از این مدل شبیه‌سازی، می‌توان با رویکرد بهینه‌سازی شبیه‌سازی به دنبال یافتن زمان مناسب برای چراغ سبز در هر جهت، به منظور کمینه‌کردن زمان انتظار بود.

مدل کلی شبیه‌سازی، با  $n$  متغیر ورودی  $(x_1, x_2, \dots, x_n)$  و  $m$  متغیر خروجی  $(y_1, y_2, \dots, y_m)$  به صورت شکل ۱-۲ زیر تعریف می‌شود.



شکل ۱-۲ مدل کلی شبیه‌سازی [۴]

فرایند یافتن بهترین ورودی‌های ممکن برای بهینه‌سازی از خروجی‌های شبیه‌سازی، بهینه‌سازی شبیه‌سازی تعریف می‌شود. خروجی‌های مدل شبیه‌سازی در فرایند بهینه‌سازی مورد استفاده قرار می‌گیرند و از پروسه جست‌وجوی بهینه برای اجزای بعدی شبیه‌سازی بازخورد تهیه می‌شود (شکل ۲-۲).



شکل ۲-۲ فرآیند بهینه‌سازی شبیه‌سازی [۴]

قالب عمومی یک مدل ریاضی بهینه‌سازی شبیه‌سازی به صورت (۱-۲) تا (۴-۲) نمایش داده می‌شود:

$$\min z = E(w.(\mathbf{X})) \quad (1-2)$$

s. t.

$$E(w_j(\mathbf{X})) \leq 0 \quad j = 1, 2, \dots, h \quad (2-2)$$

$$g_{j'}(\mathbf{X}) \leq 0 \quad j' = 1, 2, \dots, h' \quad (3-2)$$

$$x_i^l \leq x_i \leq x_i^u \quad i = 1, 2, \dots, d \quad (4-2)$$

که در آن

$\mathbf{X}$  بردار متغیرهای تصمیم مسئله،

$E(w_j(\mathbf{X}))$  امید ریاضی تابع هدف تصادفی،

$E(g_{j'}(\mathbf{X}))$  امید ریاضی توابع محدودیت تصادفی،

$g_{j'}(\mathbf{X})$  توابع محدودیت قطعی و  $x_i^l$  و  $x_i^u$  حد پایین و بالای متغیر تصمیم نام هستند.

در مسائل بهینه‌سازی شبیه‌سازی برخی توابع به صورت تصادفی هستند که همین موضوع رویکرد بهینه‌سازی شبیه‌سازی را توجیه می‌کند. در مسائل بهینه‌سازی شبیه‌سازی برخلاف سایر روش‌های بررسی مسائل برنامه‌ریزی ریاضی، همه یا تعدادی از توابع به صورت صریح موجود نیستند. این موضوع باعث عدم امکان استفاده از گرادیان محلی می‌شود. مشکل در پروسه حل این نوع مسائل عدم تخمین مناسب است که به علت ماهیت تصادفی آن در حالت تقریب گرادیان محلی است. با این حال در جایی که شکل بسته مدل ریاضی وجود ندارد، استفاده از بهینه‌سازی شبیه‌سازی برای تحلیل این نوع مسائل بسیار مفید است. با این وجود باید به علت شبیه‌سازی کامپیوتری هزینه‌ی بیش‌تر نسبت به الگوریتم‌های بهینه‌سازی را در نظر گرفت. طراحی فرآیند، طراحی سیستم‌های خدماتی، تعیین توالی عملیات، برنامه‌ریزی تولید و کنترل موجودی، طراحی بهینه در مهندسی و بهینه‌سازی پایایی از جمله مباحث مطرح در زمینه مهندسی صنایع است که توسط رویکردهای بهینه‌سازی شبیه‌سازی قابل بررسی است. در منابع [۴] و [۵] به بررسی بخشی از مساله‌های بهینه‌سازی شبیه‌سازی می‌پردازند.



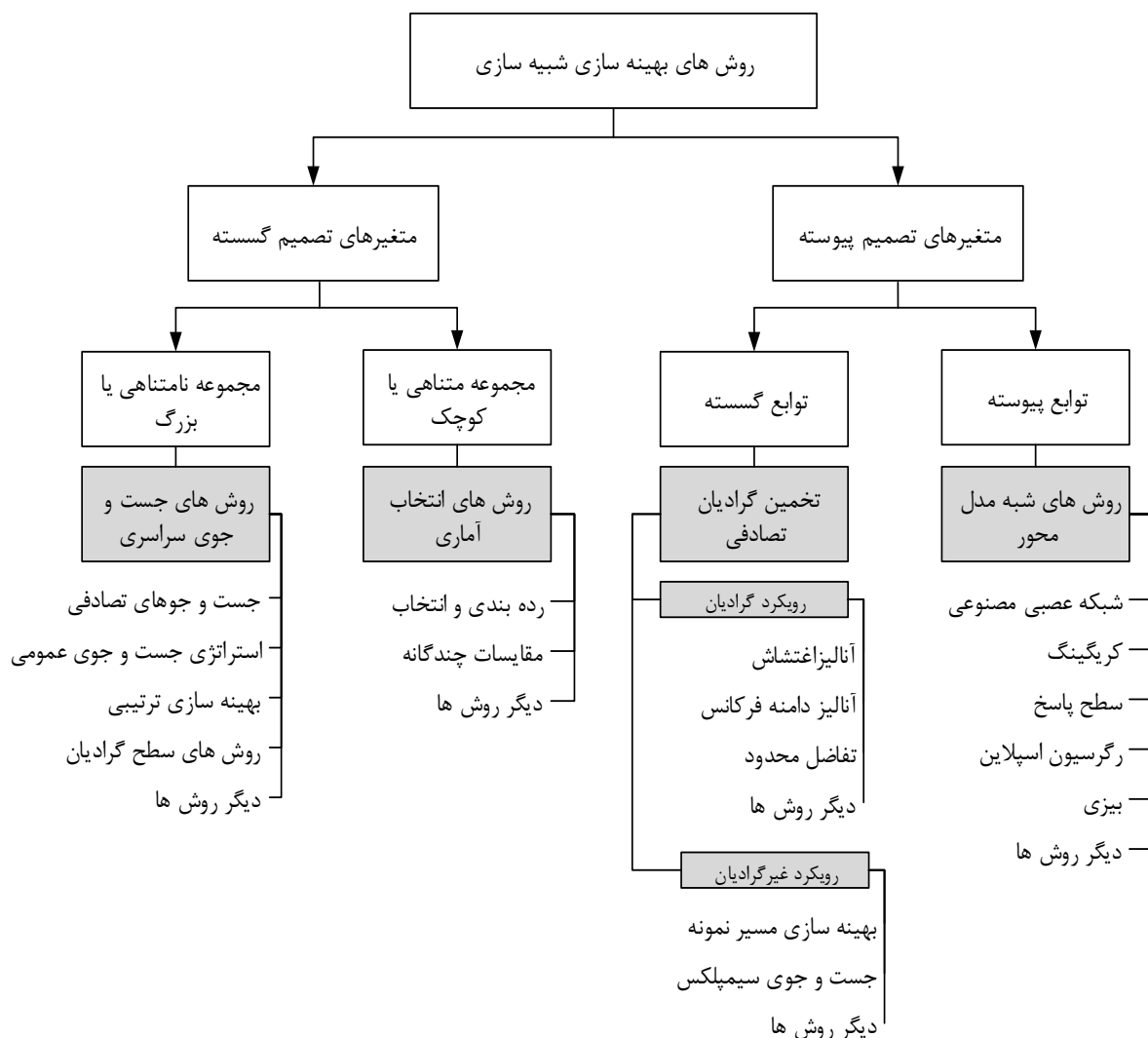
ساده ترین نوع یک م ساله بهینه سازی شبیه سازی به صورت یک م ساله تک هدفه نامقید است که متغیرهای تصمیم آن از نوع پیوسته باشند. به عنوان مثال، در یک مساله صف تک کاناله، بهینه سازی شبیه سازی به دنبال پاسخگویی به سوالات زیر است [۶]:

- چگونه زمان خدمت در یک سیستم صف تغییر یابد تا زمان انتظار مشتریان کمتر شود؟
- چه نظام صفی معیارهای عملکرد سیستم را بهبود می دهد؟
- چه ترکیبی از متغیرهای تصمیم گیری به عملکرد بهتر سیستم منجر می شود؟

## ۱-۲. دسته بندی ساختاری مدل و روش های بهینه سازی شبیه سازی

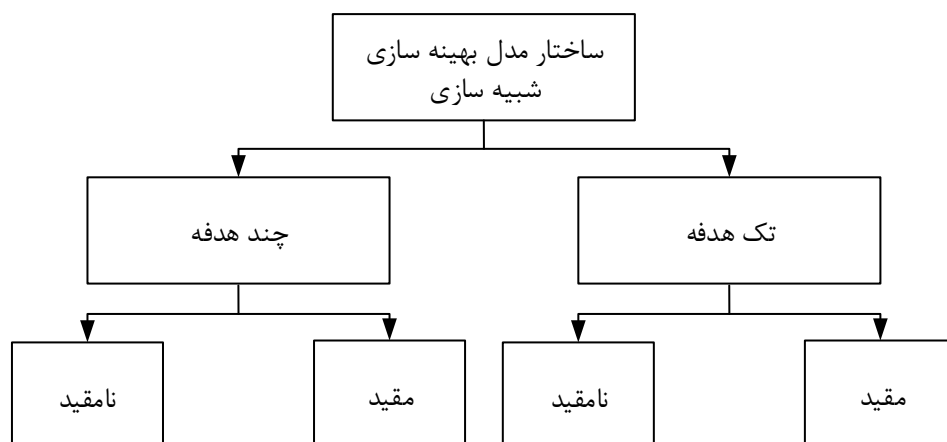
یک مساله بهینه سازی شبیه سازی از لحاظ نوع متغیر تصمیم گیری، به دو دسته کمی و کیفی قابل تقسیم بندی است. مساله بهینه سازی شبیه سازی کمی، تنها شامل متغیرهای تصمیم گیری عددی است در حالی که یک مساله بهینه سازی شبیه سازی کیفی شامل از متغیرهای وصفی و غیرکمی است. متغیرهای کیفی معمولاً به صورت صفر و یک و یا به صورت مجموعه ای از اعداد (از پیش تعیین شده) تعریف می شوند. متغیرهای کمی، به سه دسته کلی پیوسته، گسسته و آمیخته قابل تقسیم بندی هستند. مساله بهینه سازی شبیه سازی پیوسته، فقط شامل متغیرهای تصمیم گیری پیوسته، مساله بهینه سازی شبیه سازی گسسته، فقط شامل متغیرهای تصمیم گیری گسسته و یک مساله بهینه سازی شبیه سازی آمیخته شامل ترکیبی از متغیرهای پیوسته و گسسته است.

همچنین فضای تصمیم روش های بهینه سازی شبیه سازی به دو گروه گسسته و پیوسته طبقه بندی می شوند. در فضای پیوسته متغیرهای تصمیم طبق جنس توابع مورد نظر در مدل سیستم، می توان از روش های مرتبط با تخمین گرادیان و روش های بهینه سازی مبتنی بر شبه مدل ها استفاده کرد. در فضای گسسته متغیرهای تصمیم بر حسب تعداد متغیرهای روش مورد نظر انتخاب می شود. چنانچه متغیرهای تصمیم متناهی باشند روش های انتخاب آماری مانند رده بندی و انتخاب و فرایندهای مقایسات چندگانه مورد استفاده قرار می گیرند. ولی اگر متغیرهای تصمیم گسسته نامتناهی یا بزرگ بودند روش های جست و جوی سراسری مناسب می باشند. تقسیم بندی ها و روش های ذکر شده در شکل ۲-۳ قابل مشاهده است.



شکل ۲-۳ روش های بهینه سازی شبیه سازی [۷]

مسائل بهینه سازی شبیه سازی از نظر نوع تابع هدف، به دو دسته کلی تک هدفه و چند هدفه قابل تقسیم هستند. در یک مساله بهینه سازی شبیه سازی تک هدفه، بهینه سازی یک تابع هدف دارد، در صورتیکه برای مساله بهینه سازی چند هدفه، بیش از یک تابع هدف تعریف می شود. همچنین مساله بهینه سازی شبیه سازی تک هدفه یا چند هدفه، به دو دسته مقید و نامقید تقسیم بندی می شود. به این ترتیب در مسئله بهینه سازی شبیه سازی نامقید، تنها تابع هدف وجود دارد و این در حالی است که مسئله بهینه سازی شبیه سازی مقید، دست کم یک محدودیت کارکردی علاوه بر تابع هدف موجود است. تقسیم بندی ساختار مسئله های بهینه سازی [۷] در شکل ۲-۴ نمایش داده شده است.



شکل ۲-۴ تقسیم بندی مساله‌های بهینه سازی شبیه سازی از لحاظ ساختار [۱۷]

## ۲-۲. روش‌های مدل محور

یکی از روش‌های پیشگام در بهینه سازی شبیه سازی روش‌های مدل محور هستند. این روش‌ها به روش‌های حل برای مساله‌های بهینه سازی شبیه سازی پیوسته و گسسته قابل تقسیم هستند. این روش‌ها را می‌توان به صورت زیر دسته بندی کرد:

۱. روش‌های تقریب گرادیان<sup>۱</sup>
۲. روش‌های بهینه سازی تصادفی<sup>۲</sup>
۳. روش‌های آماری<sup>۳</sup>
۴. روش‌های فرا ابتکاری<sup>۴</sup>
۵. روش بهینه سازی مسیر نمونه<sup>۵</sup>

<sup>۱</sup> Gradient approximation methods  
<sup>۲</sup> Stochastic optimization methods  
<sup>۳</sup> Statistical methods  
<sup>۴</sup> Meta-heuristic methods  
<sup>۵</sup> Sample path optimization method

## ۱-۲-۲. روش‌های گرادیان محور

روش‌های تقریب گرادیان شامل روش تفاضل محدود<sup>۱</sup>، روش اغتشاش همزمان<sup>۲</sup>، روش نسبت درست‌نمایی<sup>۳</sup>، روش تحلیل اغتشاش<sup>۴</sup> و روش تحلیل حوزه فراوانی<sup>۵</sup> می‌شود. این روشها، با استفاده از مفاهیم ریاضیات پیوسته، بر مبنای تخمین گرادیان تابع هدف می‌باشند. تخمین‌های به دست آمده از گرادیان، جهت تحلیل و یا بهینه‌سازی استفاده می‌شود [۸].

## ۲-۲-۲. روش‌های بهینه‌سازی تصادفی

روش تقریب تصادفی<sup>۶</sup>، روش نیمه نیوتون تصادفی<sup>۷</sup> زیر مجموعه روش‌های بهینه‌سازی تصادفی می‌باشند. روش تقریب تصادفی [۹]، صورتی اصلاح شده از روش تندترین شیب در برنامه ریزی غیر خطی است که با استفاده از برآوردهای گرادیان در هر تکرار، سعی در کمینه‌سازی تابع هدف دارد [۱۰] و همچنین روش نیمه نیوتون تصادفی برگرفته از روش هم‌نام خود در برنامه ریزی غیرخطی است که به منظور اصلاح روش نیوتون و تضمین همگرایی آن به وجود آمده است [۸].

## ۳-۲-۲. روش‌های آماری

همچنین کاربرد روشهای آماری در مواردی است که تعداد انتخاب‌های موجود نسبتاً کم و قابل شمارش باشد. در صورتیکه سیستم قطعی باشد، این کار بسیار ساده‌تر بوده و به راحتی قابل اجرا خواهد بود. اما مقایسه طرح‌های مختلف در یک سیستم تصادفی، نیازمند تجزیه و تحلیل آماری است. روشهای متعددی برای

---

۱ Finite difference method  
۲ Simultaneous perturbation method  
۳ Likelihood ratio method  
۴ Perturbation analysis  
۵ Frequency domain analysis  
۶ Stochastic approximation  
۷ Stochastic quasi Newton method

انجام این گونه تحلیل‌های آماری وجود دارند که از جمله می‌توان به روش‌های رده بندی و انتخاب<sup>۱</sup>، انتخاب زیر مجموعه<sup>۲</sup>، ناحیه بی تفاوتی<sup>۳</sup> و مقایسات چندگانه<sup>۴</sup> اشاره کرد.

۲-۲-۴. روش های فرا ابتکاری

یافتن جواب بهینه برای بسیاری از مسائل بهینه سازی ممکن نیست. برای اینگونه مسائل معمولاً به جواب های خوب که از الگوریتم های فراابتکاری بدست آمده بسنده می شود. بر خلاف الگوریتم های بهینه سازی دقیق، فراابتکاری ها تضمینی برای بهینگی جواب های بدست آمده ارائه نمی دهند. همچنین بر خلاف الگوریتم های تقریبی، فراابتکاریها میزان نزدیکی جواب ارائه شده به جواب بهینه را مشخص نمی کنند.

در طراحی یک فراابتکاری دو معیار متناقض مد نظر قرار می گیرد: جستجوی عمومی فضای جواب<sup>۵</sup> و تمرکز بر همسایگی بهترین جواب یافته شده<sup>۶</sup>. به معیار اول جستجوی تصادفی نیز گفته می شود زیرا در هر مرحله یک جواب بصورت تصادفی از کل فضای جواب ایجاد می شود و هیچ حافظه ای مورد استفاده قرار نمی گیرد. در حالی که در exploitation که به آن جستجوی همسایگی نیز گفته می شود، در هر تکرار بهترین جواب در همسایگی جواب بهینه جاری که امکان بهبود دارد یافته می شود. با توجه به بکار گیری الگوریتم فراابتکاری بهینه سازی تجمع ذرات<sup>۷</sup> (POS) در این پژوهش، در ادامه به شرح این روش می پردازیم.

۲-۲-۴-۱. الگوریتم بهینه سازی تجمع ذرات

مبنای این روش که قابلیت جستجوی سراسری را دارد رفتار جمعی<sup>۸</sup> ذرات است که اجزا برای رسیدن به هدف نهایی بایکدیگر همکاری می کنند. این الگوریتم تقلیدیست از روش یادگیری موجودات زنده (حرکت پرنده ها و یا ماهی ها) وقتی که در یک گروه قرار دارند. برای حرکت هر ذره بر اساس راهنمای گروه و دانش شخصی هر ذره تعیین می شود و بهترین عضو راهنمایی کلی گروه را بر عهده دارد. در این روش هر ذره بیانگر

<sup>۱</sup> Ranking and selection

<sup>۲</sup> Subset selection

<sup>۳</sup> Indifference zone

<sup>۴</sup> Multiple comparisons

<sup>۵</sup> exploration

<sup>۶</sup> exploitation

<sup>۷</sup> Particle Swarm Optimization

<sup>۸</sup> Swarm behavior

یک جواب در فضای حل است که با توجه به تغییرات سرعت و مکان فضای جواب مسئله را جستجو می‌کند. عوامل تعیین سرعت عبارت اند از:

- (۱) مسیر جستجوی فعلی
- (۲) حرکت به سوی بهترین موقعیت قبلی هر ذره
- (۳) حرکت به سوی بهترین موقعیت گروه

این الگوریتم که توسط ابرهارت و کندی [۱۱] به منظور حل مسئله های برنامه ریزی غیر خطی توسعه داده شده است به منظور جهت گیری هر عنصر از سه عامل زیر تاثیر می‌گیرد:

- (۱) جایی که اکنون یک عنصر قرار دارد، می‌تواند یک جهت حرکتی باشد، زیرا شاید همین جهت مناسب باشد.
- (۲) جایی که تا بحال برای یک عنصر بهترین مکان بوده، یک جهت حرکتی برای آن عنصر است. (بهینه ذره<sup>۱</sup>)
- (۳) مکانی که تا بحال برای همه عناصر بهترین مکان بوده، یک جهت حرکتی برای هر عنصر است. (بهینه کلی<sup>۲</sup>)

اعمال تغییر سرعت بر هر یک از اجزا با رابطه (۵-۲) صورت می‌پذیرد.

$$v_i = \chi \times (v_i(t-1) + c_1 \times (pbest_i - x_i(t-1)) + c_2 \times (gbest_i - x_i(t-1))) \quad (5-2)$$

که در این رابطه (۵-۲)  $\chi$  از رابطه (۶-۲) بدست می‌آید.

$$\chi = \frac{k}{abs\left(\frac{1-\frac{c}{r}-\sqrt{abs(c^2-4 \times c)}}{2}\right)} \quad (6-2)$$

<sup>۱</sup> Personal Best  
<sup>۲</sup> Global Best

و  $k$  مقداری بین ۰ و ۱ دارد و  $C = C_1 + C_2$  است. مقدار  $C_1$  و  $C_2$  نمایانگر شدت حرکت به سمت نقطه بهینه محلی یا سراسری است.

۲-۲-۵. روش بهینه سازی مسیر نمونه

با وجود گستردگی کاربرد روشهای گرادیان محور و فرا ابتکاری، روشهای دیگری نیز در ادبیات بهینه سازی شبیه سازی وجود دارند که با بهره گیری از رویکردهایی متفاوت، به جستجوی جواب بهینه می پردازند. یکی از این روشها، بهینه سازی مسیر نمونه است. در این روش، مساله بهینه سازی شبیه سازی با استفاده از اجراهای متعدد مدل شبیه سازی، به یک مساله برنامه ریزی ریاضی تبدیل می شود و سپس بوسیله روشهای برنامه ریزی ریاضی حل می شود. در این روش ابتدا سعی در تقریب دقیق تابع هدف و سپس حل مساله قطعی حاصل می شود. در این حالت طبق (۲-۷) داریم:

$$\min f(x) = E[\psi(x, \eta)]$$

s. t.

(۲-۷)

$x \in X$

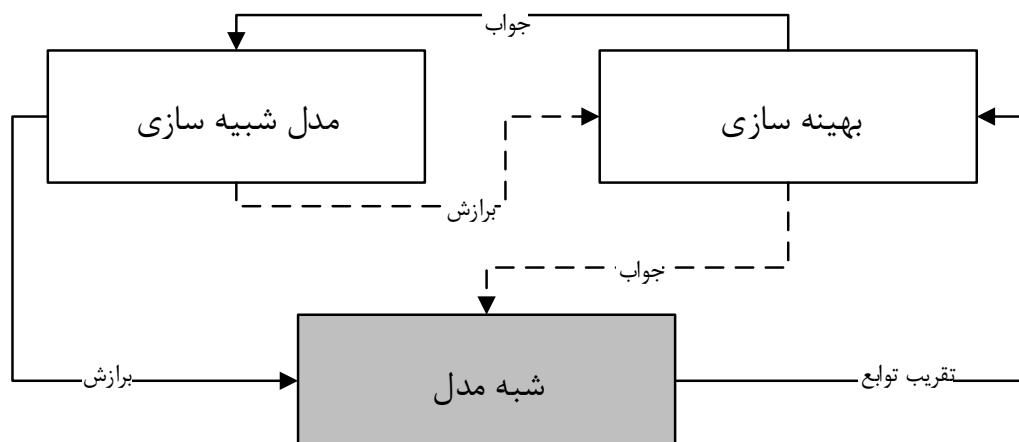
که در آن،  $\psi(x, \eta)$  متغیر تصادفی تابع عملکرد سیستم، بردار  $x$  متغیرهای ورودی و  $\eta$  یک متغیر تصادفی با تابع توزیع  $F$  می باشد. حل مساله های مقیدی که روشهای گرادیان محور قادر به حل آنها به راحتی نمی باشند از مزیتهای روش بهینه سازی مسیر نمونه است. همچنین، وجود مشتق های جزئی، تا حد زیادی در بهبود عملکرد این روش موثر خواهد بود. آندرادوتیر [۱۲] این روش را به طور دقیقتری مورد بحث قرار می دهد.

## ۳-۲. روش‌های شبه‌مدل محور

در این فصل به بررسی روش‌های شبه‌مدل محور در بهینه‌سازی شبیه‌سازی می‌پردازیم. شبه‌مدل، این امکان را به یک تحلیلگر می‌دهد تا بتواند میزان حساسیت متغیر پاسخ شبیه‌سازی، نسبت به متغیرهای ورودی را بررسی کند. هزینه زیاد اجرای مدل شبیه‌سازی اهمیت این موضوع را در بسیاری از مساله‌های دنیای واقعی برای تعیین مقدار پاسخ یک طرح توجیه می‌کند. بنابراین، به جای شبیه‌سازی یک طرح جدید، میتوان از شبه‌مدل شبیه‌سازی به منظور پیش‌بینی مقدار پاسخ آن استفاده کرد. به عبارت دیگر شبه‌مدل‌ها، توابع قطعی از متغیرهای تصمیم هستند که جهت پیش‌بینی رفتار مدل شبیه‌سازی به کار می‌روند. برآورد معیار عملکرد سیستم، نیازمند اجزای متعدد مدل شبیه‌سازی دارد. درحالی‌که با یک بار اجرای شبه‌مدل، میتوان این معیار عملکرد را با هزینه‌های بسیار کمتر برآورد کرد. یک شبه‌مدل ممکن است به عنوان دستیار مدل شبیه‌سازی نیز به کار گرفته شود. مثلاً، در مدلسازی سیستم‌های بزرگ، به جای توسعه یک مدل شبیه‌سازی، سیستم تحت بررسی به چندین زیر سیستم تفکیک میشود و مدل‌سازی زیر سیستم‌های ساده‌تر، به جای شبیه‌سازی، توسط شبه‌مدل انجام میشود. مقدار پیش‌بینی یک شبه‌مدل، مقداری قطعی از یک تابع صریح ریاضی است درحالی‌که مقدار خروجی یک مدل شبیه‌سازی، متغیری تصادفی است و ارزیابی دقیق مدل، مستلزم اجزای متعدد و زمان‌بر شبیه‌سازی است [۱۳]. با این حال، یک شبه‌مدل آریب، به خوبی قادر به شناسایی و جذب رفتار تقریبی سیستم نیست و از این حیث، یک انتخاب نامناسب در مقابل مدل شبیه‌سازی خواهد بود. عواملی همچون شکل تابعی شبه‌مدل، نوع داده‌ها و نحوه جمع‌آوری آنها، تاثیر قابل ملاحظه‌ای بر خواص آماری شبه‌مدل دارد در نتیجه توسعه یک شبه‌مدل معتبر، نیازمند بررسی دقیق عوامل تاثیرگذار بر برازندگی شبه‌مدل می‌باشد [۱۴]. مراحل توسعه یک شبه‌مدل که در شکل ۲-۵ نمایش داده شده است به شرح زیر است:

- (۱) انتخاب شکل تابعی شبه‌مدل
- (۲) انتخاب طرح آزمایش به منظور جمع‌آوری داده‌های شبیه‌سازی
- (۳) ایجاد شبه‌مدل و برآورد پارامترهای مجهول شبیه‌سازی
- (۴) اعتبار سنجی و تعیین صحت نتایج شبه‌مدل





شکل ۲-۵ بهینه‌سازی شبیه‌سازی شبه‌مدل محور [۱۳]

در مطالب بعدی به شناخت کامل‌تر انواع شبه‌مدل‌های کاربردی در بهینه‌سازی شبیه‌سازی و مفاهیم و رویکردهای آن‌ها می‌پردازیم.

### شبه مدل‌های رایج در بهینه‌سازی شبیه‌سازی

این بخش، به بررسی شبه مدل‌هایی که در سالهای اخیر، کاربرد زیادی در مباحث تحلیلی شبیه‌سازی داشته‌اند می‌پردازد. در این بخش شبه مدل رگرسیونی<sup>۱</sup>، شبه مدل شبکه عصبی مصنوعی<sup>۲</sup> و شبه مدل کرایگینگ<sup>۳</sup> بررسی خواهند شد. به دلیل استفاده از شبه‌مدل شبکه عصبی مصنوعی در این پژوهش، این نوع شبه‌مدل بیشتر مورد توجه قرار می‌گیرد. برای مطالعه بیشتر در مورد انواع شبه مدل‌ها، میتوان به منابع [۱۰] و [۱۵] مراجعه کرد.

۲-۳-۱. شبه مدل رگرسیونی

شبه مدل‌های سطح پاسخ یا شبه مدل‌های رگرسیونی، توسط باکس و ویلسن [۱۶] به منظور تقریب توابع پاسخ تصادفی توسعه داده شدند. از جمله این روش‌ها می‌توان به مدل‌های چند جمله‌ای درجه اول و دوم که

<sup>۱</sup> Regression meta-model

<sup>۲</sup> Artificial neural network meta-model

<sup>۳</sup> Kriging meta-model

بر روی متغیرهای پاسخ شبیه سازی برآزانیده میشوند اشاره داشت. یک مدل درجه اول و درجه دوم رگرسیونی به ترتیب به صورت (۸-۲) (۹-۲) نمایش داده میشوند:

$$y(x) = \beta_0 + \sum_{j=1}^d \beta_j x_j + \varepsilon \quad \varepsilon \sim NID(0, \sigma^2) \quad (8-2)$$

$$y(x) = \beta_0 + \sum_{j=1}^d \beta_j x_j + \sum_{j=1}^d \sum_{j'=1}^d \beta_{jj'} x_j x_{j'} + \varepsilon \quad \varepsilon \sim NID(0, \sigma^2) \quad (9-2)$$

که در آن،  $\varepsilon$  مولفه تصادفی مدل رگرسیون است. در مدل‌های رگرسیونی، فرض بر آن است که مولفه های تصادفی  $\varepsilon$ ، متغیرهای تصادفی مستقل دارای توزیع نرمال با میانگین صفر و واریانس یکسان اند.

بدلیل اینکه شبه مدل‌های رگرسیونی معمولاً برای تقریب نواحی کوچک از فضای آزمایش مورد استفاده قرار می‌گیرند، به آنها شبه مدل‌های محلی<sup>۱</sup> گفته می‌شود. تقریب شبه مدل‌های رگرسیونی می‌تواند در حدود کوچکی از فضای آزمایش مفید واقع شود به این دلیل که رفتار تابع پاسخ شبیه سازی در بسیاری از سیستم‌های تصادفی، پیچیده‌تر از یک رابطه خطی و یا درجه دوم است. همچنین در منبع [۱۷] شبه مدل‌های رگرسیونی در قالب طراحی آزمایشها شرح داده شده است.

## ۲-۳-۲. شبه مدل شبکه عصبی مصنوعی

شبکه عصبی مصنوعی<sup>۲</sup> یک سیستم پردازش اطلاعات است که مشخصه عملکرد آن، برگرفته از رفتار شبکه‌های عصبی بیولوژیکی است. در شبکه عصبی مصنوعی نیز با تقلید رفتاری از الگوی یادگیری از ساختار عصبی به پردازش اطلاعات پرداخته می‌شود. شبکه عصبی مصنوعی از اجزا پردازشی به نام نرون<sup>۳</sup> ایجاد شده

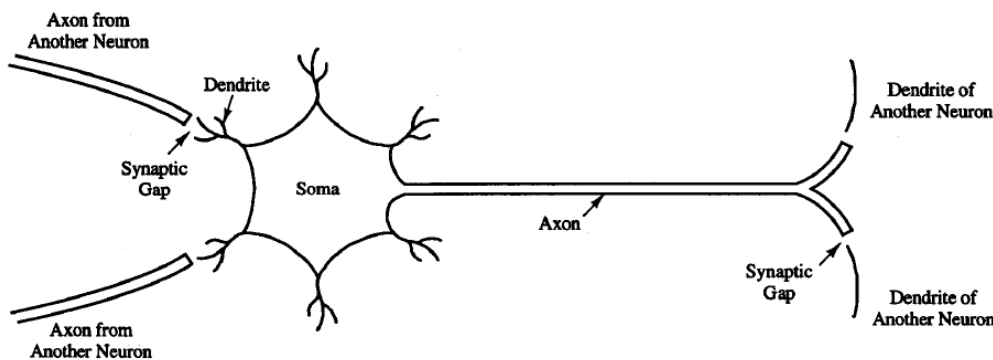
<sup>۱</sup> Local meta-model

<sup>۲</sup> Artificial Neural Network

<sup>۳</sup> Neuron

است که با هم ارتباط پیوسته و هماهنگی دارند. ساختار هر شبکه عصبی مصنوعی از تعدادی لایه تشکیل شده است که نرون‌ها به تعداد مشخص در هر لایه وجود دارند.

یک شبکه برای انجام وظیفه‌ای مشخص، مانند شناسایی الگوها و دسته‌بندی اطلاعات، در طول یک پروسه یادگیری مشابه مغز انسان تنظیم می‌شود [۱۸]. در سیستم‌های زیستی یادگیری با تنظیماتی در اتصالات سیناپسی<sup>۱</sup> که بین اعصاب قرار دارد همراه است که شبکه‌های عصبی مصنوعی هم بر همین اساس یاد می‌گیرند. در شبکه عصبی بیولوژیکی نرون‌ها توسط واحدهای ارتباطی با یکدیگر در ارتباط هستند. هر نرون را سه جزء دندریت<sup>۲</sup>، هسته<sup>۳</sup> و آکسون<sup>۴</sup> ایجاد می‌کند. سیگنال‌های ورودی توسط واحدهای دندریت دریافت می‌شوند، سپس این سیگنال‌ها به هسته نرون منتقل و در صورتیکه مجموع آن‌ها از حد معینی بیشتر شود، سیگنال‌های خروجی در هسته تولید می‌شوند و توسط آکسون به نرون‌های دیگر انتقال داده میشوند. در شکل ۲-۶ نمونه ای از یک شبکه عصبی بیولوژیکی نشان داده شده است.

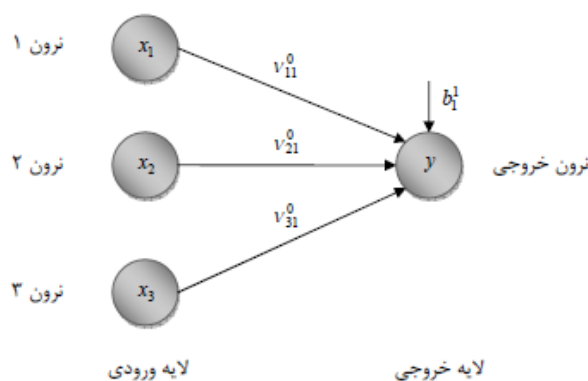


شکل ۲-۶ شبکه عصبی بیولوژیکی [۱۸]

در مرحله نخست پردازش اطلاعات در شبکه عصبی مصنوعی در واحدهای نرون انجام می‌شود. سپس سیگنال‌ها از طریق لینک‌های ارتباطی وزن‌دار میان نرون‌ها انتقال می‌یابد. سیگنال‌های ورودی هر نرون با یک تابع فعال‌سازی به سیگنال‌های خروجی تبدیل می‌شوند. این ساختار از شبکه عصبی بیولوژیکی الهام گرفته است. تابع فعال‌سازی، تابعی از داده‌های ورودی به هر نرون است. در ساختار شبکه عصبی مصنوعی تعدادی

<sup>۱</sup> Synapse  
<sup>۲</sup> Dendrite  
<sup>۳</sup> Soma  
<sup>۴</sup> Axon

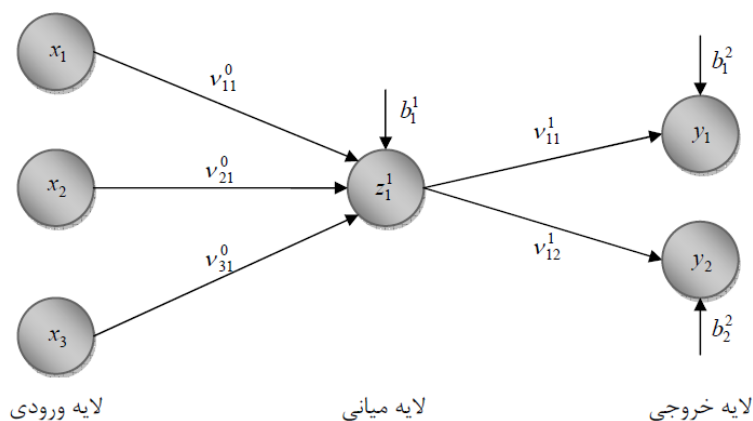
لایه مشخص وجود دارد که هر لایه متشکل از تعداد نرون‌های معین است که مطابق با نظم خاصی آرایش یافته‌اند. هر شبکه عصبی مصنوعی حتماً یک لایه ورودی و یک لایه خروجی دارد. لایه ورودی نرون‌هایی است که ورودی آن‌ها همان پارامترهای ورودی مورد نظر تحلیل‌گر یا متغیرهای تصمیم در مسائل بهینه‌سازی شبیه‌سازی هستند و در نهایت لایه خروجی مقادیر متغیرهای هدف را تولید می‌کنند. کاربرد مورد نظر جهت تقریب توابع بین متغیرهای ورودی و پاسخ جهت کمینه‌کردن میانگین مربعات خطا بین داده‌های خروجی شبیه‌سازی و داده‌های تقریب زده شده توسط شبکه برای پارامتر ورودی، وزن‌های لینک‌های ارتباطی بین نرون‌ها تشخیص داده می‌شوند و بر این اساس ساختار شبکه عصبی مصنوعی توسعه می‌یابد. شکل ۲-۷ یک شبکه عصبی مصنوعی ساده تک لایه می‌باشد.



شکل ۲-۷- شبکه عصبی مصنوعی ساده تک لایه [۱۸]

چنانچه شبکه بالا برای برازش تابع به کار رفته باشد آن‌طور که مشخص است مربوط به مسئله‌ای با دو متغیر ورودی و دو متغیر هدف و جهت شناخت رابطه‌ی ریاضی بین آن‌ها مورد استفاده قرار گرفته است. برای مثال فرض کنید از مدل شبیه‌سازی مسئله صف، به ازای نرخ ورود به سیستم و تعداد باجه‌های خدمت‌دهی به عنوان دو متغیر تصمیم، میزان انتظار افراد در صف و طول صف را به عنوان دو متغیر هدف استخراج کردیم و اکنون با استفاده از این شبکه به دنبال تقریب ارتباط بین این متغیرهای تصمیم و متغیرهای هدف مسئله هستیم. به طور معمول در معرفی تعداد لایه‌های یک شبکه عصبی لایه ورودی شمارش نمی‌شود، به همین

علت با این که سه لایه شبکه بالا را تشکیل داده است ولی آن را شبکه عصبی مصنوعی دو لایه می‌نامیم. در شکل ۲-۸ این نوع شبکه عصبی مصنوعی نمایش داده شده است.



شکل ۲-۶ شبکه عصبی مصنوعی دو لایه [۱۸]

ارتباط میان نرون‌ها در ساختار یک شبکه عصبی مصنوعی به صورت پیش‌خور و بازگشتی دسته بندی می‌شود. در شبکه عصبی مصنوعی پیش‌خور ارتباط میان نرون‌ها یک چرخه را تشکیل نمی‌دهد و این نوع شبکه عصبی اولین و ساده‌ترین نوع شبکه عصبی مصنوعی می‌باشد. همچنین زمانی شبکه عصبی مصنوعی را پس‌خور یا بازگشتی می‌نامیم که حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون‌های همان لایه یا لایه‌های قبل وجود داشته باشد.

معمولاً از سه تابع فعال‌سازی با روابط زیر برای ارتباط میان نرون‌ها استفاده می‌شود:

۱. تابع فعال‌سازی خطی<sup>۱</sup>:

<sup>۱</sup> Linear function

$$f_{lin}(x) = \sum_{i=1}^m w_i x_i + b \quad (10-2)$$

۲. تابع فعال سازی تان سیگموئید<sup>۱</sup>:

$$f_{tan}(x) = \frac{2}{1 + \exp(-2 \sum_{i=1}^m w_i x_i + b)} \quad -1 \leq x_i \leq 1 \quad (11-2)$$

۳. تابع فعال سازی لاگ سیگموئید<sup>۲</sup>:

$$f_{log}(x) = \frac{1}{1 + \exp(-(\sum_{i=1}^m w_i x_i + b))} \quad 0 \leq x_i \leq 1 \quad (12-2)$$

که  $m$  نشان دهنده تعداد نرون های لایه ی فعلی،  $w_i$  وزن لینک های ارتباطی،  $x_i$  سیگنال ورودی و  $b$  مقدار انحراف است.

برای تکمیل فرایند آموزش شبکه عصبی مصنوعی، تعیین ساختار آن و یافتن مقادیر بهینه وزن و آریبی از توابع فعال سازی استفاده می شود به نحوی که معیار عملکرد شبکه بهینه شود. معیار عملکرد تابعی از خروجی های شبکه است که برای سنجش برازندگی شبکه مورد استفاده قرار می گیرد. به عنوان مثال میانگین مربعات خطا می تواند به عنوان معیار عملکرد مورد استفاده قرار بگیرد. دو روش کلی با توجه به اطلاعات اولیه ای که در اختیار شبکه قرار می گیرد وجود دارد:

---

<sup>۱</sup> Tan sigmoid function

<sup>۲</sup> Log sigmoid function

۱. آموزش نظارت‌یافته: از طریق معیار عملکرد معین آن و زوج مرتب‌های داده شده به سیستم شامل متغیرهای ورودی و متغیرهای هدف به دست آمده از اجرای سیستم مقدار پاسخ شبکه تخمین زده خواهد شد..

۲. آموزش غیرنظارت‌یافته: متغیرهای ورودی سیستم جهت گروه‌بندی داده‌ها در اختیار شبکه قرار می‌گیرد و آموزش شبکه براساس تشابه داده‌ها در هر گروه و جداگانه صورت می‌پذیرد.

از آن‌جا که الگوهای محاسباتی شبکه عصبی مصنوعی نسبتاً جدید هستند، مزایا و کاربردهای آن در مقایسه با سایر روش‌های قدیمی‌تر کاملاً شناخته شده نیست. ولی مزایا و معایب استفاده از این روش را می‌توان در موارد زیر بیان نمود:

مزایا:

- محدودیت نداشتن در تعداد ورودی و خروجی
- هزینه کم پروسه آموزش
- رفتارهای غیرخطی و پراکنده را مدل‌سازی می‌کند.
- جهت بهبود کارایی توانایی اتصال به الگوریتم‌های بهینه‌سازی را دارد.
- حساسیت پایین نسبت به تغییرات ناگهانی
- لحاظ کردن پارامترهای گسسته

معایب:

- قابلیت تعمیم برای پیش‌بینی رفتار آینده شبکه ممکن نیست.
- برای کاربردهای اختیاری قواعدی برای آموزش شبکه وجود ندارد.

- اندازه مجموعه داده‌های مورد استفاده برای آموزش تاثیر زیادی در دقت آن دارد.
- در بعضی مسائل آموزش شبکه عصبی مصنوعی غیرممکن یا گاهی مواقع دقت مطلوبی ندارد.

### ۳-۳-۲. شبه مدل کراگینگ

شبه مدل کراگینگ توسط کرایگ در حیطه آمار جغرافیایی<sup>۱</sup> معرفی و تئوری ریاضی آن توسط مترون [۱۹] توسعه داده شد. در زمینه طراحی آزمایش‌ها، شبه مدل کرایگینگ، یک شبه مدل سراسری است که به منظور تقریب نواحی آزمایشی بزرگ مورد استفاده قرار می‌گیرد.

شبه مدل کراگینگ<sup>۲</sup> برای درونی‌یابی داده‌های طراحی آزمایش در یک محدوده بزرگ به کار می‌رود. کراگینگ همانند شبکه عصبی مصنوعی به عنوان شبه‌مدل سراسری محسوب می‌شود و برخلاف روش‌های مرسوم همسایه محور نیست. همچنین این شبه‌مدل در زمینه‌های پیش‌بینی، تحلیل حساسیت و بهینه‌سازی استوار کاربرد دارد. در حیطه بهینه‌سازی شبیه‌سازی، کراگینگ در چندین مورد توجه بسیاری از محققین قرار دارد. نمایش مدل ریاضی ساده‌ی کراگینگ به شکل (۲-۱۳) است:

$$w(\mathbf{d}) = \mu + \delta(\mathbf{d}) \quad (۲-۱۳)$$

$\mathbf{d}$  بردار متغیرهای ورودی شبیه‌سازی،  $w(\mathbf{d})$  خروجی مدل شبیه‌سازی،  $\mu$  میانگین خروجی شبیه‌سازی و  $\delta(\mathbf{d})$  مولفه تصادفی به‌دست آمده از پروسه کواریانس ایستا با میانگین صفر است.

در نتیجه شبه‌مدل کراگینگ به‌صورت (۲-۱۴) نمایش داده می‌شود:

$$y(\mathbf{d}) = \lambda(\mathbf{d}, \mathbf{D})^T w(\mathbf{D}) = \lambda^T w \quad (۲-۱۴)$$

<sup>۱</sup> Geostatistics  
<sup>۲</sup> Kriging meta-model



$\mathbf{D}$  ماتریس داده‌های ورودی،  $\mathbf{d}$  بردار ورودی پیش‌بینی شونده و  $w(\mathbf{D})$  متغیرهای پاسخ می‌باشد. در فرایند توسعه شبه‌مدل کراگینگ باید مقدار بهینه  $\lambda$  را بدست آوریم. با استفاده از بهترین برآوردگر ناریب، این مقدار به صورت (۱۵-۲) بهینه بدست می‌آید.

$$\min_{\lambda} MSE[y(\mathbf{d})] = \min_{\lambda} E[\{y(\mathbf{d}) - w(\mathbf{d})\}x^T] \quad (15-2)$$

که مقدار بهینه از رابطه (۱۶-۲) به دست می‌آید.

$$\lambda = \Gamma^{-1} \left[ \gamma + \mathbf{1} \times \left( \frac{\mathbf{1} - \mathbf{1}^T \Gamma^{-1} \gamma}{\mathbf{1}^T \Gamma^{-1} \mathbf{1}} \right) \right] \quad (16-2)$$

در اینجا  $\Gamma = cov(w_i, w'_i)$  یک ماتریس  $n \times n$  و نیمه معین مثبت است که کواریانس بین متغیرهای پاسخ از مدل شبیه‌سازی را مشخص و  $\gamma = cov(w_i, w)$  یک بردار  $n$  بعدی است که کواریانس متغیرهای پاسخ از مدل شبیه‌سازی و متغیرهای پاسخ پیش‌بینی شده را بیان می‌کند. مقدار بهینه پارامترهای شبه‌مدل کراگینگ به کمک همبستگی متغیرهای پاسخ بدست می‌آید. در این شبه‌مدل فرض می‌شود فاصله‌ی میان متغیرهای ورودی بر همبستگی میان متغیرهای پاسخ موثر است. این تابع بصورت (۱۷-۲) است.

$$\rho(w(d_i), w(d_j)) = \prod_{k=1}^K \rho(d_{ik}, d_{jk}) \quad (17-2)$$

روابطه (۱۸-۲) و (۱۹-۲) معادله شبه‌مدل کراگینگ با استفاده از روابط بالا شرح می‌دهد.

$$y(d) = \hat{\mu} + \gamma(d)^T \Gamma^{-1} (w - \hat{\mu}) \quad (18-2)$$

$$\hat{\mu} = (\mathbf{1}^T \Gamma^{-1} \mathbf{1})^{-1} \mathbf{1}^T \Gamma^{-1} w \quad (19-2)$$

منبع [۲۰] به شرح مفصل این حوزه می‌پردازد.

## ۲-۴. طراحی آزمایش

به یک سری تست که در آنها مقادیر ورودی یک سیستم بر اساس یک قانون مشخص تغییر داده می‌شود تا تغییرات در متغیر خروجی بررسی شود آزمایش گفته می‌شود. به مجموعه روش‌هایی که نحوه انجام آزمایش‌های مورد نیاز به شیوه کارا تعیین می‌کنند، طراحی آزمایش گویند و هدف از طراحی آزمایش‌ها عمدتاً شناسایی متغیرهای ورودی تاثیرگذار بر متغیر خروجی هستند. از آنجا که داده‌های مربوط به آزمایش‌ها نویزی هستند و نتایج بوسیله نویز می‌تواند تحت تاثیر قرار گیرد از روش‌های آماری مناسب جهت تجزیه و تحلیل استفاده می‌شود. در طراحی آزمایش‌ها دوباره سازی<sup>۱</sup>، تصادفی سازی<sup>۲</sup> و بلاک<sup>۳</sup> قواعد اصلی آنالیز آماری را تشکیل می‌دهد. در توسعه مدل شبیه سازی استفاده از طرح آزمایش‌های عاملی کامل<sup>۴</sup>، عاملی دو سطحی<sup>۵</sup> و مشتقات آن و طرح آزمایش فوق مکعب لاتین<sup>۶</sup> مرسومند.

### ۲-۴-۱. طرح عاملی کامل

در طرح عاملی کامل اثر همزمان دو یا چند عامل (متغیر ورودی) بر متغیر پاسخ شبیه سازی سنجیده می‌شود. در این طرح، هر عامل در چند سطر مطرح می‌شود و به ازای هر ترکیب از عوامل، مدل شبیه سازی دست کم یک بار اجرا می‌شود. نیاز زیاد این طرح به تعداد اجرای زیاد از مدل شبیه سازی مشکل طرح آزمایش کامل است.

<sup>۱</sup> replication

<sup>۲</sup> randomization

<sup>۳</sup> blocking

<sup>۴</sup> Full factorial design

<sup>۵</sup> Two-level factorial design

<sup>۶</sup> Latin hypercube design

## ۲-۴-۲. طرح عاملی دو سطحی

زمانی که تعداد عوامل مورد بررسی زیاد است، استفاده از طرح عاملی کامل بسیار زمان بر و حتی ناممکن می‌شود. در این موارد میتوان حدود فضای آزمایش را به گونه ای تنظیم نمود که تنها ۲ سطح از هر عامل مورد نیاز باشد. در این حالت، با وجود عدم پوشش کامل تابع پاسخ شبیه سازی، امکان پیش بینی تغییرات متغیر پاسخ تا مقداری وجود دارد. پس در واقع به صورت کلی این طرحها، همانند طرحهای عاملی کامل هستند اما با این تفاوت که هر عامل تنها در دو سطح مطرح می‌شود. طرحهای عاملی دو سطحی با توجه به ماهیت خود، در ایجاد شبه مدلهای محلی همانند شبه مدل های رگرسیونی درجه اول و درجه دوم به کار گرفته میشوند. همچنین روش های دیگری که از طرح عاملی دو سطحی مشتق شده باشند و نیاز به اجرای شبیه سازی کمتری در آنها باشد توسعه داده شده است. از جمله آنها می‌توان به طرح مرکب مرکزی [۲۱] طرح دو سطحی کسری [۱۷] اشاره داشت.

## ۲-۴-۳. طرح فوق مکعب لاتین

فوق مکعب لاتین (به اختصار LHS) یک طرح آزمایش سراسری است که فضای متغیرهای تصمیم را به مربع با اندازه‌های یکسان تقسیم می‌کند. محدوده‌ی هر متغیر در این روش به  $n$  بازه‌ی مساوی تقسیم می‌شود و برای هر کدام از این بازه‌ها در هر متغیر یک مقدار تصادفی درون بازه‌ای تولید می‌شود. با این کار به ازای هر متغیر  $n$  نقطه به طور پراکنده روی تمام محدوده‌ی متغیر ایجاد می‌شود [۲۲]. از ترکیب این نقاط روی تمام متغیرها، نقاط آزمایش برای اجرای شبیه‌سازی مشخص می‌شوند. بدلیل اینکه که این طرح پراکندگی نقاط را در فضای جواب را به خوبی رعایت می‌کند، برای مسائل با متغیرهای ورودی زیاد باشند، روش فوق مکعب لاتین از کارایی بالایی دارد. گام های این طرح به شرح زیر است:

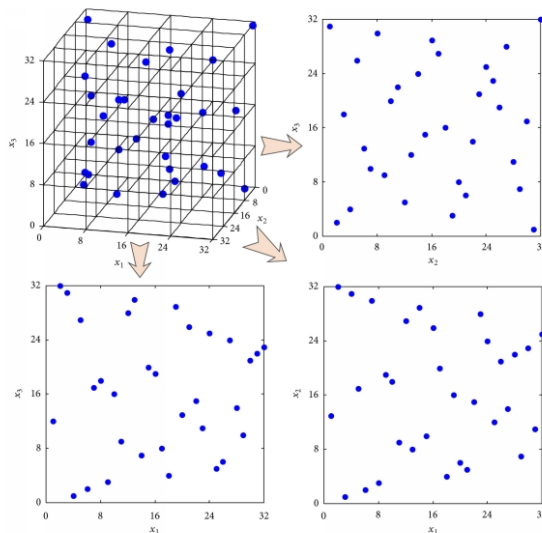
۱. تقسیم محدوده هر  $m$  متغیر تصمیم را به  $n$  بازه‌ی مساوی

۲. تولید اعداد تصادفی در هر کدام از  $n$  بازه به ازای متغیر  $j$ ام بردار  $\mathbf{d}_j = [d_{j1}, d_{j2}, \dots, d_{jn}]$   
( $j=1,2,\dots,m$ )

۳. سپس از ترکیب  $m$  بردار به دست آمده از گام دو، ماتریس نقاط آزمایش جدید ( $\mathbf{D}_{n \times m}$ ) به دست می‌آید:

$$\mathbf{D}_{n \times m} = [\mathbf{d}_1 \ \mathbf{d}_2 \ \dots \ \mathbf{d}_m] = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ \vdots & \vdots & \dots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nm} \end{bmatrix}$$

نقطه آزمایش اول  
نقطه آزمایش دوم  
 $\vdots$   
نقطه آزمایش  $n$ ام



شکل ۲-۷ نمونه ای از توزیع نقاط آزمایش در یک طرح آزمایش فوق مکعب لاتین [۲۲]

## ۲-۵. اعتبارسنجی شبه‌مدل

در این گام به بررسی این مساله پرداخته می‌شود که آیا شبه مدل فعلی نمایش مناسبی از مدل شبیه سازی است یا خیر. برای این منظور اغلب از داده های تست استفاده می‌شود. روشهای مختلفی جهت اعتبار سنجی شبه مدل ها توسعه داده شده اند از جمله آنها می‌توان به روش های زیر اشاره کرد:

روش ضریب تعیین<sup>۱</sup>

روش اعتبار سنجی متقاطع<sup>۲</sup>

روش بوتسترپ<sup>۳</sup>

۲-۵-۱. روش ضریب تعیین

این آماره فرضیات خاصی ندارد و می تواند به منظور تعیین اعتبار شبه مدل های رگرسیونی استفاده شود. ضریب  $R^2$  به صورت (۲۰-۲) تعریف می شود [۱۷].

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}(x_i) - \bar{w})^2}{\sum_{i=1}^n (\bar{w}(x_i) - \bar{w})^2} = 1 - \frac{\sum_{i=1}^n (\hat{y}(x_i) - \bar{w}(x_i))^2}{\sum_{i=1}^n (\bar{w}(x_i) - \bar{w})^2} \quad (20-2)$$

باید توجه داشت که در رابطه (۲۰-۲) با افزایش تعداد پارامترهای شبه مدل، مقدار ضریب تعیین در هر صورت افزایش می یابد. برای جلوگیری از این مشکل باید از ضریب تعیین تنظیم شده (۲۱-۲) استفاده کرد:

$$R_{adj}^2 = 1 - \frac{n-1}{n-p} (1 - R^2) \quad (21-2)$$

$P$  تعداد پارامترهای شبه مدل می باشد. برای استفاده از این رابطه باید توجه داشت که تعداد نقاط آزمایش بیشتر از تعداد پارامترهای شبه مدل باشد. در غیر این صورت، مقدار  $R_{adj}^2$  منفی خواهد شد.

<sup>۱</sup> Coefficient of determination method

<sup>۲</sup> Cross-Validation method

<sup>۳</sup> Bootstrap method

## ۲-۵-۲. روش اعتبار سنجی متقاطع

در این روش از  $p$  داده بعنوان مجموعه تست و مابقی داده‌ها بعنوان مجموعه آموزش استفاده می‌شود [۲۳]. یعنی برای  $n-p$  داده شبه مدل برازش می‌شود و با استفاده از شبه مدل برازش شده، مقدار متغیر پاسخ در  $p$  نقطه برآورد می‌شود. این کار برای کلیه حالت‌های انتخاب  $p$  عضو از کلیه اعضاء تکرار می‌شود و نتایج اعتبار سنجی بدست آمده برای آنها میانگین گرفته می‌شود. اگر  $p=1$  قرار دهیم حالت خاصی از روش بدست می‌آید که در هر تکرار یکی از  $n$  نقطه را موقت کنار گذاشته و شبه مدل مورد نظر را روی نقاط آزمایش باقیمانده برازش می‌کند. سپس با استفاده از شبه مدل برآورد شده، مقدار پاسخ این نقطه آزمایش پیش بینی می‌شود. این کار به ازای هر یک از نقاط طرح آزمایش انجام می‌شود و معیار کمی اعتبارسنجی محاسبه می‌شود.

## ۲-۵-۳. روش بوتسترپ

روش نمونه گیری مجدد یا بوتسترپ، در سال ۱۹۷۹ به عنوان یک روش کامپیوتر محور، به منظور تخمین خطای استاندارد برآوردگر معرفی شد. مشاهدات بوتسترپ، از طریق نمونه گیری با جایگذاری، از مشاهدات تصادفی مستقل و هم توزیع بدست می‌آیند [۱۷]. به عنوان نمونه اگر مجموعه اصلی به صورت  $D = \{x_1, x_2, \dots, x_n\}$  تعریف شود یک نمونه بوتسترپ، نمونه تصادفی به اندازه  $n$  از مجموعه  $D$  با احتمال انتخاب  $1/n$  برابر برای هر عضو است و به صورت مجموعه  $D^* = \{x_1^*, x_2^*, \dots, x_n^*\}$  نمایش داده می‌شود. در این حالت  $D/D^*$  بعنوان داده تست استفاده می‌شود. این فرایند  $B$  بار (معمولا ۱۰۰۰ تا ۱۰۰۰۰ بار) تکرار می‌شود و هر بار معیار کمی برازش محاسبه می‌شود و از نتایج برست آمده برای تکرارهای مختلف، میانگین گرفته می‌شود.

# فصل سوم: مرور ادبیات

### ۳. مرور ادبیات

با افزایش پیچیدگی سیستم‌ها و تمایل بیشتر جهت استفاده از شبیه سازی برای یافتن نقطه بهینه یا نزدیک به آن، بهینه سازی شبیه سازی به عنوان شاخه نسبتاً جدیدی از تحقیق در عملیات معرفی شد. در نتیجه این نگرش جدید در سال‌های اخیر مسئله‌های تصادفی پیچیده گوناگونی به صورت مدل‌های بهینه سازی شبیه سازی فرموله شده‌اند و تحقیقات وسیعی جهت حل این گونه مسائل صورت گرفته است. انتخاب روش بهینه سازی متناسب با مسئله مورد نظر امر بسیار مهمی در طراحی و پیاده سازی الگوریتم می‌باشد.

#### ۳-۱. مرور بر ادبیات بهینه‌سازی شبیه‌سازی

از اولین استفاده‌ها از بهینه سازی شبیه سازی می‌توان به داگتی و ترنسکوئیست [۲۴] در سال ۱۹۷۸ اشاره داشت که برای دستیابی به ورودی‌های بهینه شبیه‌سازی از توابع چند جمله‌ای برازش شده بر خروجی‌های شبیه سازی برای تقریب شبیه‌سازی در بین جهت‌های مناسب و اختیاری در فضای جواب استفاده نمودند. در این روش که از توابع اسپلاین استفاده شده هدف اصلی بهره‌گیری از شبه‌مدل صرفاً برای افزایش اطلاعات از مدل شبیه‌سازی بوده و در نهایت یک تابع کلی برای یافتن جهت‌های جدید جهت جستجو در تکرارهای بعدی، روی اطلاعات بدست آمده برازش می‌شود.

در سال ۱۹۸۸، آزادیور [۲۴] برای تعیین مقادیر بهینه از سیستم‌ها با متغیر تصمیم گسسته در شبیه سازی یک روش ابتکاری ارائه داد که پس از آن روش جستجوی سیمپلکس مقید براساس آن توسعه یافت. در این بدلیل تصادفی بودن متغیرهای پاسخ شبیه سازی، روش رئوس سیمپلکس مقایسه آماری شده و پس از بررسی نقاط تصمیم و رسیدن به یک مقایسه آماری مطمئن اجرای شبیه سازی متوقف جواب ارائه می‌شود.

گلین [۲۵] در سال ۱۹۸۹ جهت تقریب رفتار سراسری تابع هدف، توابع متعامد را ارزیابی نمود و نشان داد در صورت هموار بودن نسبی تابع هدف، نرخ همگرایی در حدود  $n^{-1/2}$  تعداد مشاهدات قابل حصول است. پس از آن مقاله به روش‌های یکپارچه این ایده با الگوریتم‌های بهینه‌سازی می‌پردازد.



تامپکینز و آزادیور [۲۶] در سال ۱۹۹۵ با بکار گیری ارتباط الگوریتم ژنتیک با شبیه‌سازی شیء گرا توانستند بهینه‌سازی شبیه‌سازی را بر روی متغیرهای کیفی به کار برده و پس از مقایسه آن با روش‌های عددی به نتایج خوبی را بدست آوردند. طریقه عملکرد الگوریتم به این صورت بوده که پس از مشخص شدن طراحی سیستم توسط الگوریتم ژنتیک، مدل شبیه‌سازی اجرا شده و در نهایت انتخاب‌های الگوریتم ژنتیک توسط نتایج بدست آمده هدف دار می‌شوند.

محققان زیادی از مبحث بهینه‌سازی توسط الگوریتم‌های فراابتکاری مانند الگوریتم تبرید شبیه‌سازی شده<sup>۱</sup>، الگوریتم جستجوی ممنوعه و الگوریتم ژنتیک در بهینه‌سازی شبیه‌سازی استفاده نمودند. برای مثال پاول وچانو [۲۷] با بهره‌گیری از الگوریتم ژنتیک و اصلاح آن جهت استفاده برای بهینه‌سازی شبیه‌سازی توانستند از هزینه‌های سنگین این روش که ناشی از تکرار زیاد دفعات شبیه‌سازی توسط الگوریتم است بکاهند. این استراتژی به این گونه بود که در هر نسل الگوریتم ژنتیک تنها یک فرزند تولید و با یک احتمال خاص، بهترین جواب هر نسل در ابتدای نسل بعدی تکرار می‌شود.

باشیام و فو [۲۸] روش جهت‌های شدنی که در مسائل بهینه‌سازی شبیه‌سازی پیوسته با یک محدودیت تصادفی کاربرد داشت را در مورد محدودیت‌های تصادفی در بهینه‌سازی شبیه‌سازی توسعه دادند. آن‌ها در این روش که از مطالعه بر روی یک مدل کنترل موجودی بهره می‌برد، بسیاری از فرهای اده ساز مدل را حذف کرده و مفهوم روش‌های شدنی را همراه با محدودیت‌های تصادفی مورد استفاده قرار دادند.

کاربرد روش آماری رده بندی و انتخاب و تکیب آن با الگوریتم فراابتکاری تبرید شبیه‌سازی شده را احمد و الخمیس [۲۴] برای مسئله‌های بهینه‌سازی شبیه‌سازی گسسته نامقید بکار بردند. در این روش از تعداد یک حالت توسط الگوریتم به عنوان معیار تخمین جواب بهینه استفاده شده است. از ویژگی‌های این الگوریتم احتمال زیاد آن در میل کردن به جواب بهینه سراسری که توسط مقاله نشان داده شده است می‌باشد.

وانگ [۲۹] با استفاده از عملکرد مناسب شبکه عصبی مصنوعی و کارایی و الگوریتم ژنتیک در جست‌وجوی توسعه‌ای، یک الگوریتم ترکیبی (GA-NN) بر مبنای این دو اساس ارائه داد. در این از الگوریتم شبکه عصبی مصنوعی برای پیش‌بینی مقدار تابع هدف و از الگوریتم ژنتیک جهت جست‌جوی جواب بهینه

---

<sup>۱</sup> Simulated annealing

بر اساس مقادیر برازشی استفاده شده است و توانسته است برای برخی از مسائل بهینه سازی جواب مطلوبی را بدست آورد.

الخمیس و احمد [۳۰] با استفاده از تکنیک های برنامه ریزی تصادفی یک الگوریتم تجمع ذرات را برای یک مسئله بهینه سازی شبیه سازی با محدودیت تصادفی گسسته بکار گرفتند. عمده موفقیت این الگوریتم در استفاده از روش های برنامه ریزی احتمالی در تعریف محدودیت های تصادفی است. در این الگوریتم اعمال محدودیت های مساله از طریق روش رد و قبول صورت می پذیرد.

### ۲-۳. مرور بر ادبیات بهینه سازی شبیه سازی گران قیمت

روش های بیان شده مربوط به الگوریتم های مدل محور هستند که بدلیل نیاز به تعداد زیادی از شبیه سازی در مدل شبیه سازی گران قیمت به صرفه نیستند. به همین دلیل الگوریتم های شبه مدل محور در سال اخیر مورد توجه بیشتری واقع شده اند. بهینه سازی توسط شبه مدل محور ها که بجای مفهوم تابع مدل شبیه سازی از شکل ساده تابع استفاده میکند [۳۱]. چانگ و همکاران [۳۲] متدلوژی سطح پاسخ جدیدی برای بهینه سازی شبیه سازی نامقید معرفی کردند. باییر و همکاران [۳۳] یک روش تقریب خطی محلی متوالی جهت حل مسائل بهینه سازی شبیه سازی با محدودیت های تصادفی توسعه دادند. کلینن و همکاران [۳۴] یک شبه مدل کراگینگ مناسب برای مسائل بهینه سازی شبیه سازی با محدودیت های تصادفی ارائه دادند.

اخیرا شبه مدل های شبکه عصبی مصنوعی قدرتمندی به عنوان ابزار مناسبی در برآورد مدل شبیه سازی در الگوریتم های بهینه سازی شبیه سازی بکار گرفته میشود [۳۵]. آلتیپارماک [۳۶] و همکاران برای بینه سازی اندازه بافر در سیستم خط تولید شبه مدل شبکه عصبی مصنوعی همراه با الگوریتم تبرید شبیه سازی شده توسعه دادند. وانگ [۳۷] با بکارگیری شبکه عصبی مصنوعی و الگوریتم توانست الگوریتم را برای بهینه

سازی شبیه سازی پیوسته با محدودیت بکار بگیرد. دنجیز و همکاران [۳۸] بر روی شبه مدل شبکه عصبی مصنوعی با روش آموزش جستجوی ممنوعه جهت جواب بهینه دو سیستم تولیدی تحقیق کردند. همچنین در [۳۹] شبکه عصبی مصنوعی نقش قدرتمندی در تقریب یک تابع ایفا میکند و تنها اندکی از تحقیقات به کاربرد چنین تقریب هایی در اهداف بهینه سازی میپردازند.

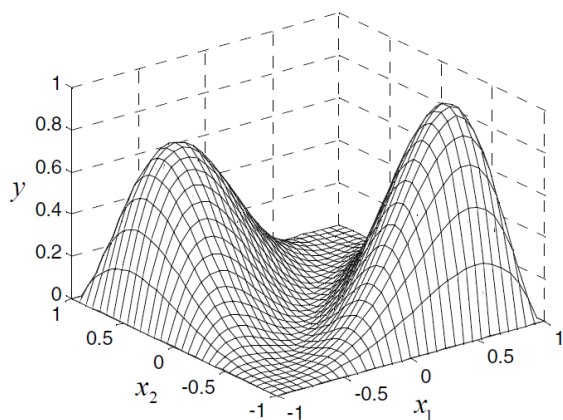
کیز و ریس [۴۰] یک الگوریتم شبه مدل محور سراسری توسعه دادند که بر پایه شبه مدل اسپلاین، روی یک طرح آزمایش متوالی ایجاد شده است. این روش، علاوه بر آنکه سعی در پوشش یکنواخت فضای آزمایش دارد، نقاط آزمایش را به قسمت هایی از فضای آزمایش اختصاص می دهد که سهم بیشتری در انحنای تابع پاسخ دارند. میزان این انحنا بر اساس علامت و مقدار مشتقات جزئی درجه دوم شبه مدل روی فضای آزمایش تعیین میشود.

در سال ۲۰۰۹ ویلمونتکس و همکاران [۴۱] روشی برجسته جهت حل مساله های بهینه سازی شبیه سازی مقید توسعه دادند. این روش با بکار گیری شبه مدل سرا سری کراگینگ در قالب یک طرح آزمایش متوالی، تابع هدف و محدودیت ها را تقریب میزند. حاصل تقریب شبه مدل معتبر در هر مرحله، یک مدل برنامه ریزی عدد صحیح غیر خطی است که با استفاده از روش انشعاب و تحدید حل می شود. پس از تخمین جواب بهینه در هر مرحله، شبه مدل به روز و فرایند بالا مجددا تکرار می شود.

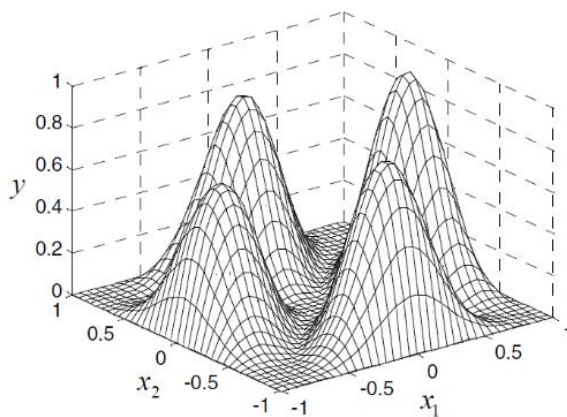
مقدم و محلوجی [۴۲] با ترکیب دو رویکرد مدل محور و شبه مدل محور الگوریتمی جدید و دو مرحله ای را جهت حل مسائل بهینه سازی شبیه سازی نیمه گران قیمت ارائه دادند که علاوه بر بهره گیری از ویژگی های رویکرد شبه مدل محور، از خطای برازش آن با صرف زمان کمتری می کاهد. لازم به ذکر است که در الگوریتم پیشنهادی برای استفاده از ویژگی های الگوریتم شبه مدل محور از شبه مدل کراگینگ و برای بهینه سازی مدل شبیه سازی و شبه مدل برازش شده از الگوریتم بهینه سازی تجمع ذرات استفاده شده است.

### ۳-۳. مرور ادبیات مقایسه شبه مدل کراگینگ و شبکه عصبی مصنوعی

در میان شبه‌مدل‌های بهینه‌سازی شبیه‌سازی شبکه عصبی و کراگینگ جذاب‌ترین روش‌های تخمین فضای جواب است که برخلاف روش‌های رگرسیون، توانایی تخمین فضای جواب با یک مدل کلی را دارند. شبکه عصبی با توانایی تقریب توابع غیر خطی نا شناخته کاربرد بسیار وسیعی در زمینه‌های گوناگون دارد. همچنین شبه‌مدل کراگینگ که در واقع روش رگرسیون پارامتری رابطه فضایی است، درونیایی نسبتاً دقیقی را ارائه می‌دهد. اگرچه هر دو روش شبه‌مدل مناسبی را ارائه می‌کند، عدم وجود تست‌های عددی کمی و کیفی کاربرد عملی آنها را محدود می‌کند. به همین علت تعداد محدودی محقق با توسعه تعدادی معیار عملکرد به مقایسه این دو شبه‌مدل پرداخته‌اند. از جمله این محققان می‌توان به [۴۳] و یارنیو و بای‌گوانچن [۴۴] اشاره کرد که با تعریف چهار معیار عملکرد مناسب جهت برآورد عملکرد شبکه عصبی و کراگینگ، به مقایسه این دو شبه‌مدل پرداختند. این معیارها برای هر دو شبه‌مدل ۱۰۰ بار برای توابع (۱-۳) و (۲-۳) که سطح پاسخ حقیقی آنها در شکل ۱-۳ و ۲-۳ نمایش داده شده است اجرا شد و نتیجه حاصل مورد بحث قرار گرفت.



شکل ۱-۳ سطح پاسخ تابع  $f_1(x)$  [۴۳]



شکل ۲-۳ سطح پاسخ تابع  $f_2(x)$  [۴۴]

$$f_1(x) = 3639.35(1 - x_1^2)(1 - x_2^2)(4 + x_1)[0.99x_1 - 0.99x_2]^2 \quad (1-3)$$

$$f_2(x) = 0.704376(1 - x_1^2)(1 - x_2^2)(4 + x_1)[0.05x_1^2 - 0.05x_2^2]^2 \quad (2-3)$$

نتیجه این آزمایش به این صورت مشخص شده است که (۱) در صورت وجود نمونه داده های کافی هر دو تقریب شبکه عصبی و کراگینگ ناحیه صحیح مقدار بهینه را تعیین می کنند. (۲) در تعداد نقطه آزمایش برابر، نقطه بهینه ارائه شده توسط شبه مدل روش کراگینگ دقیق تر می باشد. (۳) اختلاف معناداری در تعیین ویژگی های مهم فضای جواب در این دو شبه مدل وجود ندارد. البته این نتایج محدود به آزمایش بر روی دو تابع بوده و نمی توان به نتیجه حتمی رسید. در ضمن این احتمال وجود دارد با تنظیم بیشتر پارامترهای شبکه عصبی به جواب های بهتری می رسیدیم. یکی از مهمترین عوامل عملکرد شبکه عصبی، تعداد نرون های لایه های پنهانی می باشد. برای تعیین این مقدار، بایستی فاکتور های زیادی را در نظر گرفت. ابعاد فضای جواب، ویژگی های تابع مورد نظر جهت مدل کردن، تعداد نقاط آزمایش و احتمال بیش برزش<sup>۱</sup> و کم برزش<sup>۲</sup> از مهمترین این فاکتورها می باشد. با وجود اینکه قانون کلی برای تعیین کردن مقدار نرون های لایه های پنهانی تا به حال بدست نیامده است، چندین محقق در روش های گوناگون به روابط کاربردی متعددی رسیده اند. برای نمونه در منبع [۴۴] یان به این موضوع پرداخته است و در تحقیقی جداگانه ریگونی [۴۵] رابطه (۳-۳) را با  $n$  متغیر ورودی،  $m$  مقدار خروجی  $h$  نرون لایه پنهان تعیین نموده است. در اینجا برای جلوگیری از بیش برزش پارامتر  $l$  را با رابطه (۳-۴) که در توسط ضرب تعداد داده های آموزش  $q$  در تعداد متغیر خروجی  $m$  تعیین می کنیم.

$$l = h(n + m + 1) + m \quad (3-3)$$

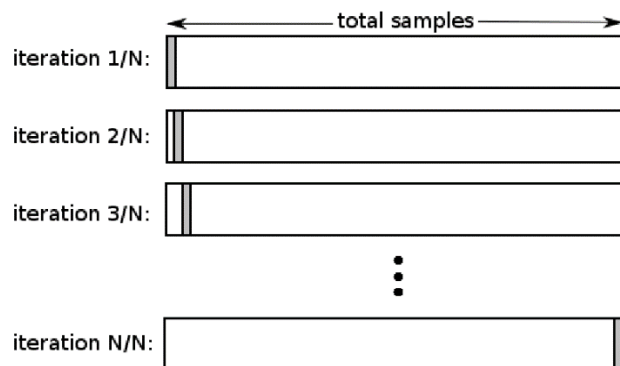
$$l < q \times m \quad (4-3)$$

پس از برزش شبکه عصبی مصنوعی مرحله اعتبار سنجی شبه مدل بایستی صورت گیرد. بررسی های متعددی توسط میکشیکر [۴۶] و کرسی [۴۷] در این رابطه صورت پذیرفته است. به دلیل عدم نیاز به شبیه سازی اضافه تر نقاط که منجر به جلوگیری از شبیه سازی مجدد مدل شبیه سازی می شود، این روش در اعتبار سنجی شبه مدل شبکه عصبی مصنوعی و کراگینگ بسیار متداول می باشد. در شکل ۳-۳ فرایند

---

<sup>۱</sup> overfitting  
<sup>۲</sup> underfitting

انتخاب و محاسبه اعتبار شبه مدل به روش اعتبار سنجی متقاطع کنارگذاری تکی<sup>۱</sup> نمایش داده شده است [۲۴].



شکل ۳-۳ نمایش روش اعتبار سنجی متقاطع کنارگذاری تکی [۴۶]

در این روش در هر مرحله یک نقطه  $x_i$  از سایر نقاط طراحی کنار گذاشته می شود و شبه مدل برای سایر نقاط برازش می شود. جهت محاسبه خطای برازش، شبه مدل برازش شده برای ارزیابی سطح پاسخ متغیر حذف شده استفاده می شود. به این روش اعتبار سنجی متقاطع، کنارگذاری تکی گفته می شود. برای بررسی بیشتر این روش می توان به منبع [۲۴] مراجعه کرد.

<sup>۱</sup> Leave-one-out cross validation

## **فصل چهارم :**

**الگوریتم بهینه‌سازی شبه‌مدل محور**

**پیشنهادی**

## ۴. الگوریتم بهینه‌سازی شبه‌مدل محور پیشنهادی

الگوریتم پیشنهادی در این پژوهش جهت حل مسائل بهینه‌سازی شبیه‌سازی نیمه‌گران قیمت توسعه داده شده است. در واقع این الگوریتم برای بهینه‌سازی مدل‌های شبیه‌سازی مفید خواهد بود که طول زمان اجرای هر دور شبیه‌سازی آن‌ها بین ۲ تا ۵ دقیقه است.

با وجود اینکه خروجی مدل شبیه‌سازی از آنچه که توسط یک شبه‌مدل برآورد می‌شود بسیار دقیق‌تر می‌باشد، در روش‌های شبه‌مدل خروجی‌های شبیه‌سازی تنها برای برازش شبه‌مدل مورد استفاده قرار می‌گیرند. در حقیقت روش‌های شبه‌مدل محور مراحل برازش شبه‌مدل و بهینه‌سازی به صورت مجزا عمل می‌کنند به طوری که پس از برازش شبه‌مدل و اجرای مراحل اعتبار‌سنجی، گام‌های بهینه‌سازی شروع می‌شوند و در نتیجه خروجی‌های شبیه‌سازی در گام‌های بهینه‌سازی استفاده نمی‌شود و تنها به شبه‌مدل برازش شده اتکا می‌شود. یکی از ایده‌های این الگوریتم استفاده از نقاط شبیه‌سازی شده برای افزایش سرعت و دقت بهینه‌سازیست.

نکته قابل توجه دیگر این است که پس از تشخیص نامعتبر بودن شبه‌مدل، در هر تکرار یک نقطه شبیه‌سازی دیگر به سایر نقاط جهت افزایش دقت شبه‌مدل افزوده می‌شود. این روزه برای شبیه‌سازی‌های بسیار گران قیمت طراحی شده است در صورتی که زمان محاسبات مراحل اعتبار‌سنجی در مقابل زمان شبیه‌سازی‌های نسبتاً گران قابل توجه می‌باشد. این درحالی است که با رد شدن مجدد شبه‌مدل در هر تکرار به اضافه شدن نقطه شبیه‌سازی شده نیاز است و در این صورت این امکان وجود دارد که الگوریتم در این حلقه گیر کند. پس در نتیجه در مسائلی که شبیه‌سازی آن‌ها نیمه‌گران است اضافه نمودن چند نقطه به صرفه‌تر از یک نقطه است.

باتوجه به مطالب گفته شده الگوریتم حاضر از هر دو ویژگی روش‌های مدل محور و شبه‌مدل محور بهره می‌گیرد. ای الگوریتم از دو فاز تشکیل شده است. در فاز اول الگوریتم به صورت یک الگوریتم مدل محور از شبیه‌سازی برای ارزیابی نقاط استفاده می‌کند. در همین راستا پس از اینکه شبه‌مدل معتبر تشخیص داده شد، فاز دوم شروع می‌شود و با جایگذاری شبه‌مدل بدست آمده با مدل شبیه‌سازی، الگوریتم به صورت یک الگوریتم شبه‌مدل محور عمل می‌کند.



همچنین با توجه به ویژگی‌های الگوریتم بهینه‌سازی تجمع ذرات مبتنی بر جمعیت محور بودن و عدم حذف و اضافه شدن نقاط جدید در طول مراحل بهینه‌سازی (به این صورت که اجزا تغییر نمیکنند و تنها به سمت بهینه حرکت می‌نمایند) در الگوریتم با کمی تفاوت در فاز اول و دوم از آن استفاده شده است.

در منبع [۴۲] این الگوریتم با تکیه بر روش شبه مدل کراگینگ برای اولین بار معرفی شد و نتایج حاصل از اجرای آن بر روی توابع آزمایشی<sup>۱</sup> نشان داده شده است. در ادامه علاوه بر بهره‌گیری از شبه‌مدل شبکه عصبی مصنوعی به عنوان شبه مدل این الگوریتم به مقایسه نتایج حاصل از توابع آزمایشی بر روی این الگوریتم با الگوریتم اولیه می‌پردازیم.

الگوریتم از دو فاز کلی تشکیل شده است. فاز اول شامل ۶ گام و فاز دوم ۳ گام دارد. بنابه توضیحات صورت گرفته الگوریتم به صورت شکل ۴-۱ نمایش داده می‌شود. در ادامه به تشریح گام‌های الگوریتم پرداخته خواهد شد.

#### ۴-۱. تعریف متغیرهای مسئله (فاز اول - گام ۱)

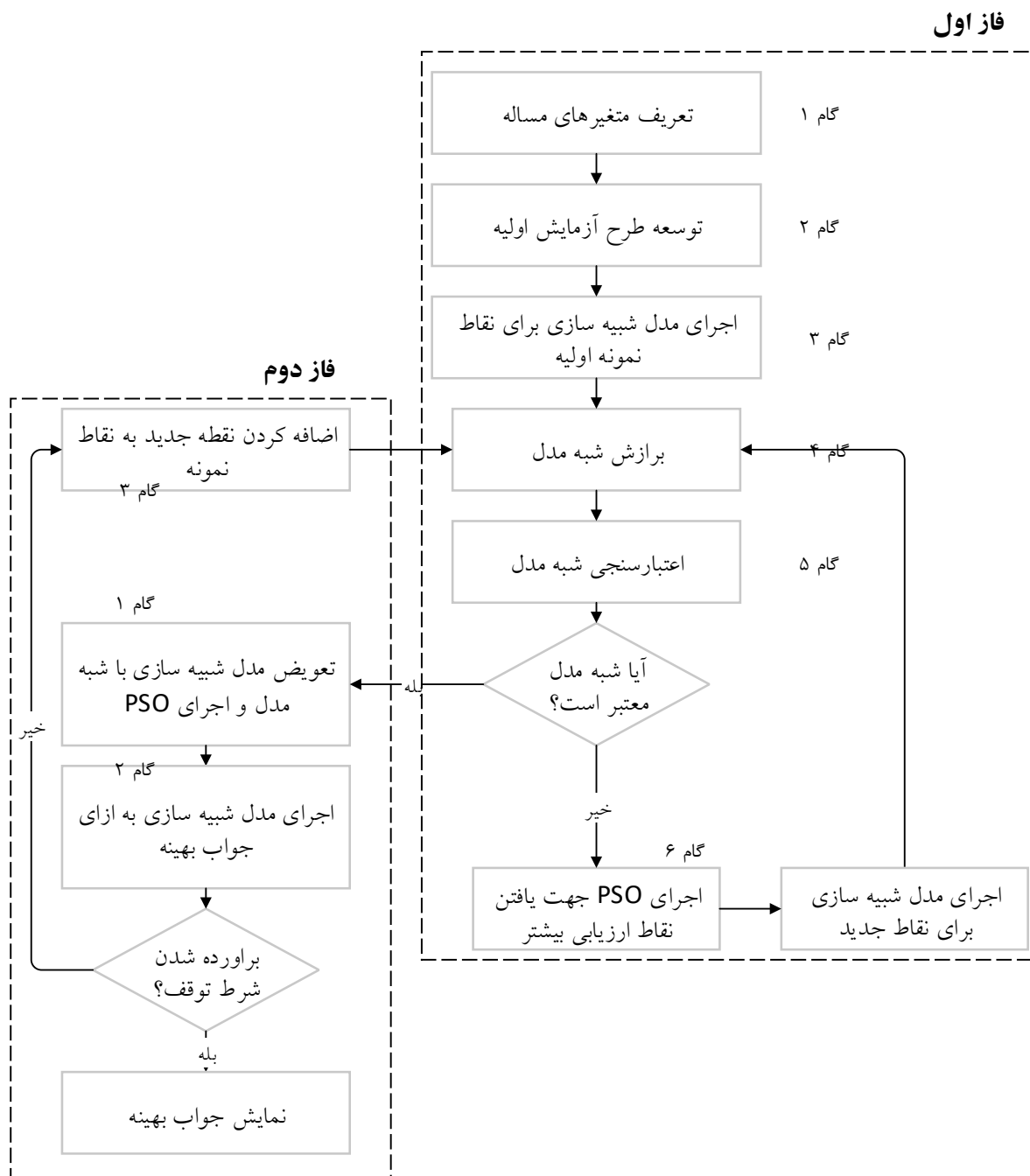
اولین مرحله تعریف متغیر  $x$ ، و بازه آنها  $[l_j, u_j]$ ، به ازای  $d, \dots, 2, 1, i$ ، و همچنین متغیر پاسخ  $w(x)$  که در آن  $x$  بردار متغیرهای تصمیم می‌باشد. علاوه بر این به پارامترهای دیگر همچون  $m$  اندازه جمعیت،  $I$  حداکثر تکرار الگوریتم بهینه‌سازی و همچنین  $r$  تعداد دوباره‌سازی شبه‌سازی<sup>۲</sup> مورد نیاز می‌باشد.

#### ۴-۲. توسعه طرح آزمایش اولیه (فاز اول - گام ۲)

روش معمول جهت طرح آزمایش اولیه برای شبکه عصبی مصنوعی فوق مکعب لاتین است که یک طرح آزمایش سراسریست و بر این اساس است که فضای متغیرهای تصمیم را به مربع با اندازه‌های یکسان تقسیم می‌کند. برتری و دلیل اصلی استفاده این روش در برازش شبه مدل شبکه عصبی این است که حداقل فاصله

<sup>۱</sup> Test function  
<sup>۲</sup> Simulation replication

بین نقاط را حداکثر می‌کند. باید توجه داشت که تعداد نقاط اولیه برابر با تعداد جمعیت ذرات در الگوریتم بهینه سازی جمعیت ذرات است.



شکل ۴-۱ الگوریتم بهینه‌سازی شبیه‌سازی‌های نیمه گران قیمت

#### ۳-۴. اجرای مدل شبیه سازی برای نقاط نمونه اولیه (فاز اول - گام ۳)

در گام سوم الگوریتم مدل شبیه سازی  $I$  مرتبه به ازای تمامی نقاط طراحی اولیه اجرا شده و  $\bar{w}(x_i), i = 1, \dots, m$  میانگین خروجی شبیه سازی ها به عنوان متغیرهای سطح پاسخ در نظر گرفته می شود.

#### ۴-۴. برازش شبه مدل شبکه عصبی (فاز اول - گام ۴)

با استفاده از خروجی های شبیه سازی  $\bar{w}(x_i), i = 1, \dots, m$  شبه مدل شبکه عصبی بر روی داده ها برازش می شود. روش مناسب جهت کمینه کردن میانگین مربعات خطای برازش، شبکه عصبی پیش خور (feedforward) دولایه که است با روش تکرار شونده با تعیین وزن ها به آموزش شبکه می پردازد. در این نوع شبکه جهت برازش در لایه پنهان از تابع فعال سازی تانسیگموئید و در لایه خروجی از تابع خطی استفاده می شود. در آموزش شبکه عصبی پیش خور جهت برازش خم و برآورد اوزان و آریبی توابع فعال سازی از الگوریتم پس انتشار لونبرگ-مارکارد<sup>۱</sup> به دلیل همگرایی سریع تر در آموزش شبکه های با اندازه متوسط و مسائل کاربردی نسبت به روش پس انتشار استفاده می کنیم. این روش که از محاسبه ماتریس هیسین<sup>۲</sup> اجتناب می کند برای آموزش شبکه پیش خور با تخمین متعدد مورد نیاز صدها وزن، بسیار دقیق عمل می کند.

همان طور که در بخش ۳-۳ گفته شد به غیر از مسائلی خاص، هنوز تئوری اثبات شده ای جهت تعیین ساختار بهینه شبکه عصبی مصنوعی برای مسائل عمومی توسعه داده نشده است. با این وجود در مرحله ساخت شبکه عصبی مصنوعی باید توجه داشت در صورتی که شبکه اولیه را کوچک در نظر بگیریم رابطه مناسب بین ورودی ها و خروجی ها بدست نمی آید و در غیر این صورت اگر شبکه اولیه را بزرگ در نظر بگیریم، بیش برازش بوجود آمده قدرت تخمین کلی شبکه را به شدت کاهش می دهد.

<sup>۱</sup> Levenberg-Marquardt backpropagation

<sup>۲</sup> Hessian

#### ۵-۴. اعتبار سنجی شبه‌مدل (فاز اول - گام ۵)

همان طور که در بخش ۳-۳ گفته شد به دلیل عدم نیاز به شبه سازی مجدد در روش اعتبار سنجی متقاطع کارگذاری تکی از این روش استفاده می‌کنیم. مقدار انحراف خروجی شبیه سازی و خروجی ارائه شده توسط شبه مدل به ازای کنار گذاری  $\hat{t}$  به صورت آماره  $t$  از رابطه (۱-۴) به دست می‌آید

$$t_{r-1}^i = \frac{\bar{w}(x_i) - \bar{y}^*(x_i)}{\sqrt{\text{var}(\bar{w}(x_i)) + \text{var}(y^*(x_i))}} \quad (1-4)$$

که  $\bar{w}(x_i)$  متوسط خروجی شبیه‌سازی بوتسترپ و  $\bar{y}^*(x_i)$  متوسط خروجی‌های شبه‌مدل شبکه عصبی در حالتی است که  $x_i$  را از شبه‌مدل کنار گذاشته باشیم.  $\text{var}(\bar{w}(x_i))$  و  $\text{var}(y^*(x_i))$  به ترتیب واریانس متوسط خروجی شبیه‌سازی بوتسترپ و واریانس بوتسترپ شبه مدل شبکه عصبی  $x_i$  می‌باشند. بعد از بدست آوردن تمامی ترکیبات آماره  $t_{r-1}^i$  را بدست می‌آوریم و اگر این مقدار بنا به آزمون فرض معنادار باشد، شبه‌مدل شبکه عصبی رد می‌شود و به گام ۶ می‌رویم.

#### ۶-۴. اجرای مدل بهینه‌سازی برای یافتن نقاط جدید (فاز اول - گام ۶)

در صورت رد شبه‌مدل بر اثر آزمون فرض، الگوریتم از الگوریتم بهینه سازی جهت ارزیابی جواب‌های بعدی استفاده می‌کند. نقاط ارزیابی شده جدید به نقاط طراحی قبلی می‌پیوندند و شبه مدل جدید برازش می‌گردد تا شبه‌مدل ارتقا یابد و این حلقه تا جایی که اعتبار سنجی شبه‌مدل از طریق آزمون فرض مورد تایید باشد ادامه می‌یابد. سپس فاز دوم الگوریتم آغاز می‌شود.

#### ۷-۴. تعویض مدل شبیه سازی با شبه مدل (فاز دوم)

در فاز دوم مدل شبیه سازی جای خود را به شبه مدل معتبر داده تا در مراحل بهینه سازی سرعت همگرایی به جواب بهینه افزایش یابد. در انتهای الگوریتم برای اطمینان از اعتبار جواب نهایی، جواب بدست

آمده شبیه سازی می شود و در صورتی که شرط توقف برآورده شده بود، جواب بدست آمده همان جواب بهینه است و الگوریتم متوقف خواهد شد. در غیر این صورت، یک نقطه شبیه سازی شده به مجموعه نقاط طراحی اضافه شده و شبه مدل دوباره برازش خواهد شد. این الگوریتم تا جایی که تکرارها به مقدار مشخصی برسند متوقف خواهد شد.

#### ۴-۸. تفاوت ویژگی های الگوریتم بهینه سازی تجمع ذرات مورد استفاده در فاز اول و دوم

دلیل اصلی استفاده از الگوریتم بهینه سازی تجمع ذرات علاوه بر مبنای روش که برای حل مسائل غیر خطی توسعه داده شده، ساده و قابل کدنویسی بودن آن است. همچنین این الگوریتم به تعداد پارامترهای کمی وابسته است. در بسیاری از منابع برتری الگوریتم PSO بر سایر الگوریتم های فرا ابتکاری از برخی از جنبه ها مورد بحث قرار گرفته است. ساهین و آکی [۴۸] با استفاده از تعریف تابع فاصله و در نظر گرفتن معیار های همگرایی و زمان چندین الگوریتم فرا ابتکاری را مورد بررسی قرار دادند. با وجود تغییر قدرت این الگوریتم ها در همگرایی به دلیل حساس بودن به تنظیم پارامترهای اولیه، با توجه به تنظیم معیارهای معرفی شده الگوریتم PSO را کاراترین الگوریتم در محدوده این معیارها معرفی نمودند.

با توجه به تفاوت ماهیت بهینه سازی فاز اول دوم و بکار گیری الگوریتم بهینه سازی تجمع ذرات در هر دو فاز، تغییراتی بر روی الگوریتم اعمال شده است. برای بکار گیری این الگوریتم  $k$  برابر ۱ و به دلیل اینکه مقدار  $C_1$  بایستی از ۴ بیشتر شود، مقدار  $C_1$  و  $C_2$  هر کدام ۲,۰۵ در نظر گرفته شده اند.

در فاز نخست الگوریتم برای دسترسی به جواب از مدل شبیه سازی استفاده شده است. به همین دلیل برای جلوگیری از افزایش تعداد شبیه سازی مجدد، در گام بهینه سازی مربوط به این فاز از جستجوی محلی استفاده نشده است. در صورتی که در فاز دوم به دلیل اینکه جهت ارزیابی نقاط مورد نظر از شبه مدل استفاده می کنیم، در گام بهینه سازی آن از جستجوی محلی مورد استفاده قرار گرفته است.

# **فصل پنجم :**

## **بررسی عددی**

## ۵. بررسی عددی

در این فصل با اجرای توابع آزمایشی<sup>۱</sup> رایج موجود در ادبیات موضوع [۴۹] به اجرای الگوریتم معرفی شده در فصل چهارم می‌پردازیم. از جمله این توابع آزمایش می‌توان به اسفیر<sup>۲</sup>، گریوانک<sup>۳</sup> [۵۰]، راسترژین<sup>۴</sup> [۵۱] و سایر توابع آزمایش مانند اسچافر<sup>۵</sup> و رسنبراک<sup>۶</sup> که برای آشنایی بیشتر میتوان به منبع [۵۲] مراجعه کرد. الگوریتم را برای هر یک از توابع آزمایش ۲۰ بار به صورت مستقل اجرا کرده و از میانگین و انحراف معیار جواب‌های بدست آمده برای مقایسه استفاده خواهند شد. الگوریتم در نرم افزار متلب R۲۰۱۶a کد شده است و نتایج محاسباتی از نوت بوک دو هسته ای با پردازشگر ۲ گیگاهرتزی و حافظه اصلی ۶ گیگابایت بدست آمده است. همچنین در فرایند آموزش شبکه عصبی مصنوعی، از جعبه ابزار شبکه عصبی<sup>۷</sup> متلب با انتخاب الگوریتم آموزش دهنده لونبرگ مارکوارت استفاده شده است.

با اجرای متعدد الگوریتم مقدار اولیه پارامترهای ثابت الگوریتم طبق جدول ۵-۱ بدست آمده است.

جدول ۵-۱ راه اندازی مقدار اولیه پارامترها

مقدار	توضیح پارامتر
۱۰	تعداد نقاط آزمایش اولیه
۱	تعداد تکرار شبیه‌سازی در هر نقطه
۰,۰۰۰۱	حداکثر میانگین مربع خطای مورد قبول
۱۰	حداکثر تعداد تکرار الگوریتم مدل محور
۶۰	حداکثر تعداد تکرار الگوریتم شبه مدل محور
۰,۱	سطح معنادار بودن آزمون اعتبار سنجی
۲	حداکثر تعداد لایه شبکه عصبی مصنوعی
۱۰	تعداد نقاط طراحی اضافه شده در هر تکرار الگوریتم مدل محور

<sup>۱</sup> Test function

<sup>۲</sup> Sphere

<sup>۳</sup> Griewank

<sup>۴</sup> Rastrigin

<sup>۵</sup> Schafre

<sup>۶</sup> Rosenbrock

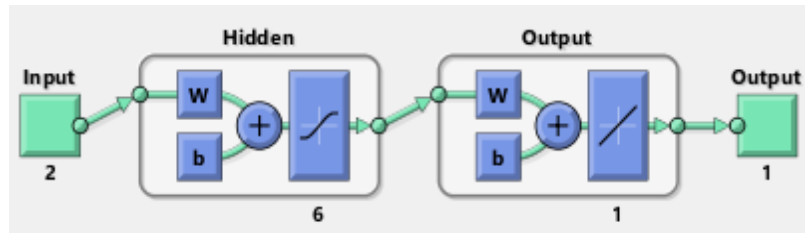
<sup>۷</sup> Neural network toolbox

۱-۵. اجرای الگوریتم برای تابع اسفیر:

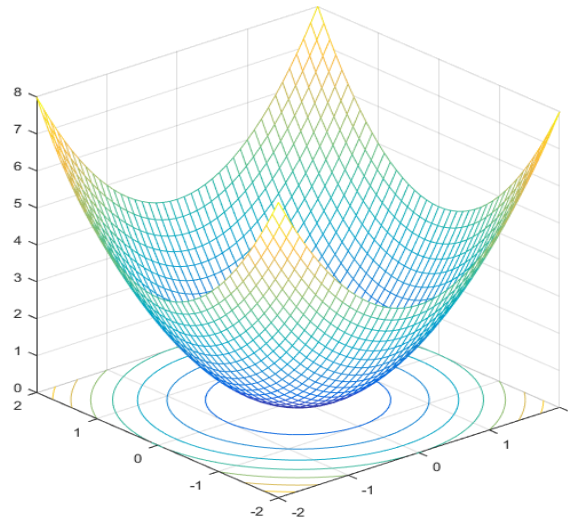
شکل کلی تابع آزمایش اسفیر به صورت (۱-۵) است.

$$f(x) = \sum_{j=1}^d x_j^2 \quad d = 2, -2 \leq x_j \leq 2, j = 1, 2 \quad (1-5)$$

مقدار بهینه این تابع  $f^* = 0$  است که در نقطه  $x^* = [0, 0]$  روی می‌دهد. شکل ۱-۵ ساختار شبکه عصبی مصنوعی و شکل ۳-۵ سطح پاسخ و خطوط تراز سطح پاسخ واقعی تابع را نمایش می‌دهد.



شکل ۱-۵ ساختار شبکه عصبی مصنوعی برازش شده



شکل ۲-۵ سطح پاسخ واقعی تابع



با توجه به اندازه فضای جواب و دو متغیر ورودی و یک متغیر پاسخ تابع اسفیر، با استفاده از اجزای متعدد، اندازه مناسب شبکه ۶ نرون در لایه پنهان بدست آمده است. نتایج بدست آمده از ۱۰ بار اجرای الگوریتم در جدول ۲-۵ ثبت شده و بهترین جواب با علامت \* مشخص شده است.

جدول ۲-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع اسفیر

شماره اجرا	جواب بهینه $x^* = [x_1, x_2]$	مقدار بهینه $f^*$
۱	$[-0.0411, -0.0450]$	۰,۰۰۳۷
۲	$[0.0016, 0.0067]$	۰,۰۱۵
۳	$[-0.0127, 0.0002]$	۰,۰۰۰۱۶ *
۴	$[0.0367, -0.0031]$	۰,۰۰۱۳۵
۵	$[-0.0025, -0.0775]$	۰,۰۰۰۶
۶	$[0.0173, -0.0667]$	۰,۰۰۰۴۷
۷	$[-0.0168, 0.0002]$	۰,۰۰۰۰۲۸
۸	$[-0.102, 0.173]$	۰,۰۰۴۰۵
۹	$[0.0285, -0.2073]$	۰,۰۰۴۳۸
۱۰	$[-0.263, 0.0179]$	۰,۰۰۶۹۴
		میانگین
		انحراف معیار
		۰,۰۱۸۵
		۰,۰۲۴۲

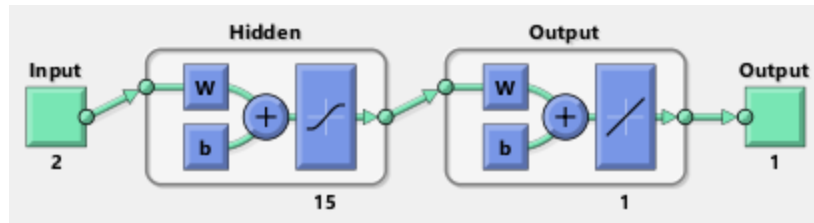
۲-۵. اجرای الگوریتم برای تابع گریوانک:

شکل کلی تابع آزمایش گریوانک به صورت (۲-۵) می باشد.

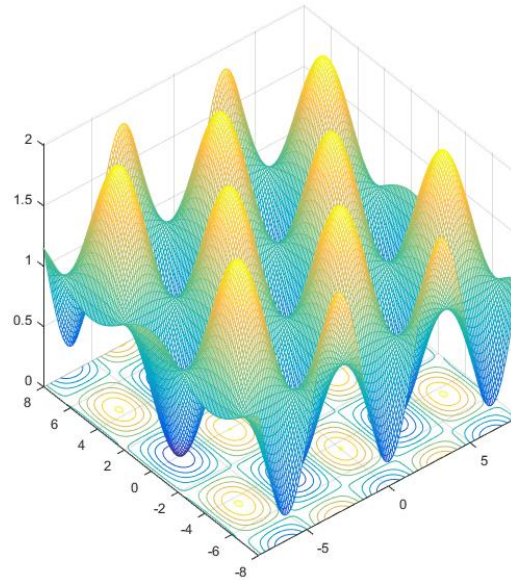
$$f(x) = \frac{1}{4 \dots} \sum_{j=1}^d x_j^2 - \prod_{j=1}^d \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1 \quad (2-5)$$

$$d = 2. -1 \leq x_j \leq 1. j = 1, 2$$

مقدار بهینه این تابع  $f^* = 0$  می باشد که در نقطه  $x^* = [0, 0]$  روی می دهد. شکل ۳-۵ ساختار شبکه عصبی مصنوعی و شکل ۴-۵ سطح پاسخ و خطوط تراز سطح پاسخ واقعی تابع را نمایش می دهد.



شکل ۳-۵ ساختار شبکه عصبی مصنوعی برازش شده



شکل ۴-۵ سطح پاسخ واقعی تابع

با توجه به اندازه فضای جواب و دو متغیر ورودی و یک متغیر پاسخ تابع گریوانک ، با استفاده از اجراهای متعدد، اندازه مناسب شبکه ۱۵ نرون در لایه پنهان بدست آمده است. نتایج بدست آمده از ۱۰ بار اجرای الگوریتم در جدول ۳-۵ ثبت شده و بهترین جواب با علامت \* مشخص شده است.

جدول ۳-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع گریوانک

مقدار بهینه $f^*$	جواب بهینه $x^* = [x_1, x_2]$	شماره اجرا
۰.۰۰۷	[۳.۱۷۳۵, -۴.۴۳۳۹]	۱
۰.۰۴۷۹	[-۳.۲۶۷۵, -۴.۰۷۶۲]	۲
۰.۰۴۵۹	[-۳.۴۰۲۳, ۴.۵۷۲۸]	۳
۰.۰۲۰۹	[۶.۴۲۸۷, ۰.۰۱۵۳]	۴
۰.۰۱۴	[-۶.۳۷۵۱, -۰.۰۲۶۰]	۵

۰.۰۰۰۰۷*	[-۰.۰۱۱۸, -۰.۰۰۲۸]	۶
۰.۰۲۳۴	[-۶.۱۸۰, ۰.۱۸۶]	۷
۰.۰۴۱۹	[-۶.۳۵۰۵, ۰.۳۴۵۳]	۸
۰.۰۰۹۳	[-۳.۱۹۹, ۴.۴۵۷]	۹
۰.۰۳۸	[-۶.۱۲۷, ۰.۲۶۳]	۱۰
۰.۰۲۴۸		میانگین
۰.۰۱۷۵		انحراف معیار

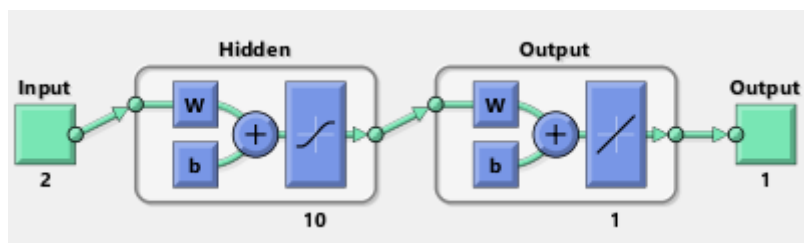
۳-۵. اجرای الگوریتم برای تابع اسپچافر اف ۶:

شکل کلی تابع آزمایش اسپچافر اف ۶ به صورت (۳-۵) می باشد.

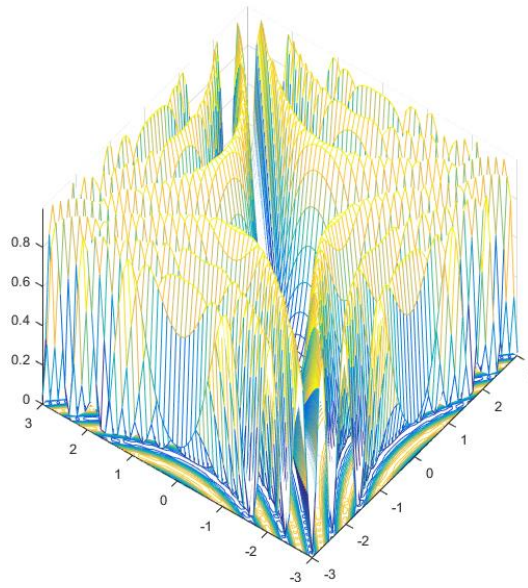
$$f(x) = 0.5 - \frac{(\sin \sqrt{(x_1^2 + x_2^2)})^2}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (3-5)$$

$$d = 2. -3 \leq x_j \leq 3. j = 1, 2$$

مقدار بهینه این تابع  $f^* = 0$  است که در نقطه  $x^* = [0, 0]$  روی می دهد. شکل ۵-۵ ساختار شبکه عصبی مصنوعی و شکل ۶-۵ سطح پاسخ و خطوط تراز سطح پاسخ واقعی تابع را نمایش می دهد.



شکل ۵-۵ ساختار شبکه عصبی مصنوعی برازش شده



شکل ۴-۵ سطح پاسخ واقعی تابع

با توجه به اندازه فضای جواب و دو متغیر ورودی و یک متغیر پاسخ تابع اسپاگر، با استفاده از اجراهای متعدد، اندازه مناسب شبکه ۱۰ نرون در لایه پنهان بدست آمده است. نتایج بدست آمده از ۱۰ بار اجرای الگوریتم در جدول ۴-۵ ثبت شده است.

جدول ۴-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع اسپاگر

شماره اجرا	جواب بهینه $x^* = [x_1, x_2]$	مقدار بهینه $f^*$
۱	$[-0.0026, -0.0018]$	۰.۰۰۰۰
۲	$[0.0186, -0.0181]$	۰.۰۰۰۰
۳	$[0.0168, 0.021]$	۰.۰۰۰۰
۴	$[0.0065, 0.0069]$	۰.۰۰۰۰
۵	$[0.0039, 0.0037]$	۰.۰۰۰۰
۶	$[-0.0155, 0.0985]$	۰.۰۰۰۰
۷	$[-0.0029, 0.003]$	۰.۰۰۰۰
۸	$[-0.0075, 0.0015]$	۰.۰۰۰۰
۹	$[0.0255, -0.0219]$	۰.۰۰۰۰
۱۰	$[0.0004, 0.0025]$	۰.۰۰۰۰
	میانگین	۰
	انحراف معیار	۰

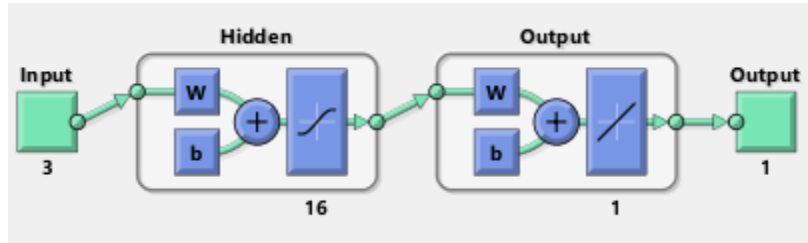
۴-۵. اجرای الگوریتم برای تابع راسترچین:

شکل کلی تابع آزمایش راسترچین به صورت (۴-۵) می‌باشد.

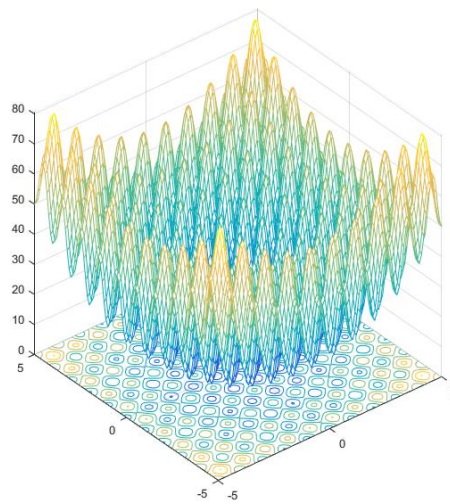
$$d = 3. -5 \leq x_j \leq 5. j = 1, 2, 3$$

$$f(x) = \sum_{j=1}^d (x_j^2 - 10 \cos(2 \times \pi x_j) + 10) \quad (4-5)$$

مقدار بهینه این تابع  $f^* = 0$  است که در نقطه  $x^* = [0 \dots 0]$  روی می‌دهد. شکل ۷-۵ ساختار شبکه عصبی مصنوعی و شکل ۸-۵ سطح پاسخ و خطوط تراز سطح پاسخ واقعی تابع در حالت دارا بودن دو متغیر ورودی را نمایش می‌دهد.



شکل ۷-۵ ساختار شبکه عصبی مصنوعی برازش شده



شکل ۸-۵ سطح پاسخ واقعی تابع در حالت دو بعدی

با توجه به اندازه فضای جواب و دو متغیر ورودی و یک متغیر پاسخ تابع راسترین ، با استفاده از اجراهای متعدد، اندازه مناسب شبکه ۱۶ نرون در لایه پنهان بدست آمده است. نتایج بدست آمده از ۱۰ بار اجرای الگوریتم در جدول ۵-۵ ثبت شده است.

جدول ۵-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع راسترین

شماره اجرا	جواب بهینه $x^* = [x_1, x_2, x_3]$	مقدار بهینه $f^*$
۱	$[-۰.۹۲۷, ۰.۹۶۰, ۰.۹۵۹]$	۴.۳۵۹
۲	$[۰.۹۸۱, -۰.۰۶۷, -۱.۰۸۸]$	۴.۶۲۰
۳	$[-۰.۰۳۶, -۰.۱۱۱, ۰.۱۱۵]$	۵.۱۳۸
۴	$[-۰.۹۶۰, ۱.۰۱۵, ۰.۰۲۶]$	۲.۴۴۹*
۵	$[۰.۹۳۱۸, ۰.۹۸۶, -۰.۰۷۴]$	۳.۸۵۷
۶	$[-۱.۹۷۹, -۰.۰۶۲, -۰.۹۷۴]$	۵.۸۳۵
۷	$[۰.۸۸۷, ۰.۹۵۶, -۰.۰۶۹]$	۵.۴۲۷
۸	$[-۰.۹۲۶, -۰.۹۹۴, ۰.۹۹۳]$	۳.۹۰۵
۹	$[-۰.۰۲۶, -۱.۰۳۹, ۱.۱۰۷]$	۴.۹۳۳
۱۰	$[۰.۰۰۳, ۱.۰۴۳, ۱.۰۲۵]$	۲.۶۴۷
میانگین		۴.۳۱۷
انحراف معیار		۱.۱۲۲

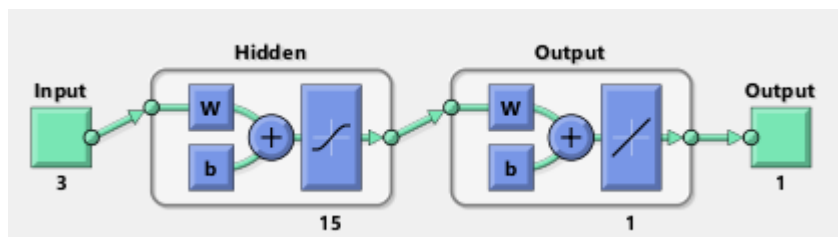
### ۵-۵. اجرای الگوریتم برای تابع رسنبراک:

شکل کلی تابع آزمایش رسنبراک به صورت (۵-۵) می باشد.

$$d = ۳. -۳ \leq x_j \leq ۳. j = ۱, ۲, ۳$$

$$f(x) = \sum_{j=1}^{d-1} \left( ۱۰ \cdot (x_{j+1} - x_j^2) \right)^2 + (x_j - ۱)^2 \quad (۵-۵)$$

مقدار بهینه این تابع  $f^* = ۰$  است که در نقطه  $x^* = [۱, ۱, ۱]$  روی می دهد. شکل ۵-۹ ساختار شبکه عصبی مصنوعی را نمایش می دهد. همچنین به دلیل اینکه این تابع سه متغیر ورودی دارد، شکل سه بعدی قابلیت ترسیم ندارد.



شکل ۹-۵ ساختار شبکه عصبی مصنوعی برازش شده

با توجه به اندازه فضای جواب و دو متغیر ورودی و یک متغیر پاسخ تابع رسنبراک ، با استفاده از اجزای متعدد، اندازه مناسب شبکه ۱۵ نرون در لایه پنهان بدست آمده است. نتایج بدست آمده از ۱۰ بار اجرای الگوریتم در جدول ۶-۵ ثبت شده است.

جدول ۶-۵ نتایج بدست آمده از اجرای الگوریتم برای تابع رسنبراک

مقدار بهینه $f^*$	جواب بهینه $x^* = [x_1, x_2, x_3]$	شماره اجرا
۰.۰۵۹۱	[۱.۰۳۸۶, ۱.۱۰۲۷, ۱.۰۲۳۳]	۱
۰.۱۵۹	[۱.۳۶۷۸, ۱.۸۸۶۲, -۲.۸۲۱۶]	۲
۰.۴۷۹	[۰.۹۰۷۷, ۰.۷۵۵۳, ۰.۷۴۸۸]	۳
۰.۱۶۵	[۰.۹۰۰, ۰.۷۷۲, -۰.۰۸۳]	۴
۰.۱۶۴	[۱.۲۲۵, ۱.۵۳۵, ۰.۳۹۱]	۵
۰.۰۷۲	[۰.۸۲۸, ۰.۷۰۷, ۲.۷۱۱]	۶
۰.۰۰۴	[۱.۰۵۴, ۱.۱۰۹, ۲.۸۸۱]	۷
۰.۰۰۰۳*	[۰.۹۸۹۱, ۰.۹۷۶, ۲.۶۹۲]	۸
۰.۰۲۴	[۱.۱۴۹, ۱.۳۱۶, ۰.۷۱۵]	۹
۰.۲۱	[۱.۱۸۳, ۱.۳۵۷, ۰.۸۴۰۱]	۱۰
۰.۱۳۳۶	میانگین	
۰.۱۴۲۸	انحراف معیار	

## ۵-۶. مقایسه بکار گیری شبه مدل شبکه عصبی مصنوعی در برابر شبه مدل کراگینگ

طبق مطالب گفته شده الگوریتم معرفی شده نسبت به الگوریتم شبه مدل محور به تعداد بیشتری اجرای شبیه سازی نیاز دارد. طبق آزمایشات صورت گرفته این تعداد تقریباً برابر است با  $m \times r/2$  که در آن  $m$  تعداد نقاط اضافه شده در هر تکرار الگوریتم و  $r$  تعداد دفعات دوباره سازی شبیه سازی برای هر نقطه است. در صورتی که الگوریتم به میزان تقریبی  $(m-1)/m$  زمان اعتبار سنجی شبه مدل را کاهش می‌دهد. با این حساب در صورتی که  $T_v$  زمان اعتبار سنجی شبه مدل و  $T_s$  زمان یکبار اجرای شبیه سازی باشد، اگر برای مساله بهینه سازی رابطه  $(mr/2)T_s > ((m-1)/m)T_v$  برقرار باشد، الگوریتم معرفی شده از نظر زمان محاسباتی و میزان کیفیت جواب بهینه بهتر عمل می‌کند. در حقیقت طبق مطالب گذشته هدف این الگوریتم برای مسائل شبیه سازی نیمه گرانی است که اجرای یکبار شبیه سازی آن زمان زیادی طول نکشد.

جدول ۵-۷ نمایشگر خلاصه نتایج بدست آمده از مقایسه کیفیت جواب بهینه حاصل از بکارگیری شبه مدل شبکه عصبی مصنوعی در برابر بکارگیری شبه مدل کراگینگ در الگوریتم معرفی شده است.

جدول ۵-۷ نتایج عددی

روش کراگینگ		روش شبکه عصبی		مقدار جواب بهینه	نام تابع
انحراف معیار	میانگین	انحراف معیار	میانگین		
۰,۰۹	۰,۱۸۹	۰,۰۲۴	۰,۰۱۸	۰	اسفیر
۰,۲۲	۰,۳۳۵	۰,۰۱۷	۰,۰۲۴	۰	گریوانک
۰,۰۵	۰,۵۰۱	۰	۰	۰	اسچافر
۰,۱۶	۰,۷۷۲	۱,۱۲۲	۴,۳۱۷	۰	راسترچین
۱,۶۷	۳,۱۰۳	۰,۱۴۲	۰,۱۳۳	۰	رسنبراک



طبق جدول ۵-۷ توانایی الگوریتم برای یافتن جواب بهتر با بکار گیری شبکه عصبی مصنوعی برای ۴ تابع آزمایش از ۵ تابع آزمایش افزایش می‌یابد. به همین جهت به دلیل سادگی بکار گیری هر دو روش شبه مدل، توصیه می‌شود از هر دو روش برازش شبه مدل در الگوریتم برای یافتن جواب بهینه استفاده شود. برای مقایسه بیشتر این دو روش برازش شبه مدل می‌توان به منبع [۴۴] مراجعه کرد.

برای مقایسه کارایی این دو روش از روی داده های بدست آمده از آنالیز واریانس<sup>۱</sup> استفاده می‌کنیم. برای مشاهده روش مقایسه می‌توان به منبع [۵۳] مراجعه کرد. در این آزمایش با توجه به اینکه آماره  $F$  برابر با ۱,۰۰ و مقدار  $P$ -value برابر ۰,۳۲۱ بدست آمد فرض صفر رد نمی‌شود که بیانگر عدم تفاوت آماری روش شبکه عصبی مصنوعی نسبت به روش کراگینگ به ازای این پنج تابع آزمایش است. جدول تحلیل واریانس مقایسه این دو روش در جدول ۵-۸ قابل مشاهده می‌باشد.

جدول ۵-۸ تحلیل واریانس

منبع	مجموع مربعات خطا	درجه آزادی	میانگین مربعات خطا	آماره $F$	$p$ -value
تیمارها	۵۹۵۷۷	۱	۵۹۵۷۷	۱,۰۰	۰,۳۲۱
بلاک‌ها	۲۴۷۴۳۸	۴	۶۱۸۶۰	۱,۰۳	۰,۳۹۴
خطا	۵۶۲۶۲۹۸	۹۴	۵۹۸۵۴	۱,۰۴	۰,۳۹۲
مجموع	۵۹۳۳۳۱۳	۹۹	-	-	-

<sup>۱</sup> ANOVA

## **فصل ششم :**

# **نتیجه‌گیری و تعیین پژوهش‌های آتی**

## ۶. نتیجه‌گیری و تعیین پژوهش‌های آتی

بهینه‌سازی شبیه‌سازی به عنوان ابزار تحلیل و تصمیم‌گیری در حوزه پژوهش عملیاتیست و برای مسائلی که شکل ریاضی سیستم بسیار پیچیده و غیر قابل دسترس است بسیار پرکاربرد است. در این پژوهش ابتدا به شناخت بهینه‌سازی شبیه‌سازی و تکنیک‌های مورد استفاده و دسته‌بندی‌های مربوطه در آن پرداختیم. روش کلی مدل شبیه‌سازی و راه‌های بهینه‌سازی آن را بیان کردیم و سپس روش‌های مر سوم در ادبیات برای طرح آزمایش و برازش توابع توسط شبه‌مدل‌ها معرفی شدند.

همان‌طور که در فصل‌های قبل عنوان کردیم، الگوریتم‌های گوناگونی برای تعیین مقدار بهینه شبیه‌سازی‌ها ارائه شده است که به طور کلی جهت بهینه‌سازی‌های مسائل با شبیه‌سازی‌های طولانی و گران و یا شبیه‌سازی‌های با زمان کم مورد استفاده قرار می‌گیرند. در فصل چهارم این پژوهش علاوه بر تشریح الگوریتم بهینه‌سازی شبیه‌سازی‌های نیمه‌گران قیمت، با بکارگیری شبه‌مدل شبکه عصبی مصنوعی در فصل پنجم به مقایسه الگوریتم جدید با الگوریتم پایه پرداختیم.

ایده کلی الگوریتم کاهش زمان بهینه‌سازی مسائل با هزینه محاسباتی متوسط از طریق افزایش نقاط طراحی در هر مرحله بهینه‌سازیست. این الگوریتم در ابتدا بعد از تعریف متغیرهای مسئله و توسعه طرح آزمایش نقاط طراحی را شبیه‌سازی می‌کند و سپس در گام ۴ شبه‌مدل را برازش می‌کند. در صورتی که اعتبار مدل تایید نشود الگوریتم بهینه‌سازی تجمع ذرات با نقاط بیشتر شروع به جستجوی سراسری فضای جواب برای حرکت ذرات به سمت نقطه بهینه می‌کند و نقاط جدیدی را تولید می‌کند. با شبیه‌سازی نقاط جدید به برازش مجدد شبه‌مدل می‌پردازیم. در صورتی که اعتبار شبه‌مدل رد شد این حلقه را آنقدر ادامه خواهیم داد که اعتبار شبه‌مدل تایید شود. در این لحظه تعویض شبه‌مدل به جای مدل شبیه‌سازی فاز دوم الگوریتم شروع می‌شود و با جستجوی محلی نقطه بهینه را پیدا خواهیم کرد. در نهایت اختلاف این نقطه با مقدار واقعی شبیه‌سازی اندازه‌گیری خواهد شد و در صورت برآورده کردن انتظار این نقطه به عنوان نقطه بهینه انتخاب خواهد شد و در غیر این صورت با اضافه کردن یک نقطه به نقاط طراحی شبه‌مدل را مجدد برازش می‌کنیم تا در نهایت به شرط توقف برسیم.

الگوریتم در نرم افزار متلب ۲۰۱۶ کد نویسی شده و برای ارزیابی عملکرد الگوریتم و مقایسه آن با الگوریتم پایه از پنج تابع آزمایش استفاده شد. در ۴ تابع از ۵ تابع آزمایش از بکار گیری شبه مدل شبکه عصبی نتیجه بهتری حاصل شد و در نهایت توسط آزمون تحلیل واریانس فرض برتری شبه مدل شبکه عصبی مصنوعی بر شبه مدل کراگینگ رد شد. ریشه این عدم تفاوت آماری بیشتر در انتخاب نوع تابع و تعداد و محدوده پارامتر هاست که در بخش ۳-۳ به آن پرداختیم.

الگوریتم ترکیبی مدل محور و شبه مدل محور ارائه شده در این پژوهش، علی رغم آنکه از عملکرد خوبی برخوردارند، اما میزان توانایی آن را می توان از چندین جنبه مورد بررسی قرارداد و در صورت امکان آن را بهبود بخشید. مقدار این بهبود را می توان بر اساس دو معیار زمان محاسبات و کیفیت جواب نهایی مورد ارزیابی قرارداد. در ادامه به مواردی از مطالعات آتی در زمینه بهبود عملکرد الگوریتم ترکیبی پیشنهادی اشاره می شود.

(۱) در بخش پنجم با بیان روابط ریاضی در مورد زمان شبیه سازی هایی که مناسب این الگوریتم هستند پرداخته شد. اجرای متعدد الگوریتم برای بهینه سازی های متعدد به کمک مدل های شبیه سازی سیستم های واقعی که زمان شبیه سازی آنها در بازه تعیین شده این الگوریتم هستند و مقایسه آنها با سایر الگوریتم های بهینه سازی می تواند این بازه را در مورد شبیه سازی های واقعی تعیین کند.

(۲) همان طور که اشاره شد الگوریتم بهینه سازی تجمع ذرات در دو فاز الگوریتم با ویژگی های مجزا مورد استفاده قرار گرفته است. بدلیل هدف متفاوت این دو فاز در گام بهینه سازی، تعریف صحیح ویژگی رفتار جستجوی این الگوریتم در این دو فاز بسیار مهم است. با بهبود روش یافتن نقاط جدید در فضا و همچنین تغییرات در نوع جستجو، این انتظار وجود دارد که سرعت همگرایی الگوریتم معرفی شده به سمت جواب بهینه افزایش یابد.

(۳) در قسمت برازش شبه مدل این پژوهش از شبکه عصبی مصنوعی و در مقاله پایه از شبه مدل کراگینگ استفاده شده است. استفاده از سایر شبه مدل ها مانند عملکرد پایه رادیال<sup>۱</sup> و یا شبه مدل های بیزین<sup>۲</sup> که با کمک توزیع پیشین موجود که در واقع دانش قبلی تحلیلگر است و به

---

<sup>۱</sup> Radial Basis Function

<sup>۲</sup> Bayesian

پیش‌بینی تابع هدف می‌پردازد، ممکن است موجب بهبود الگوریتم به صورت کلی ویا برای مسائل خاص شود.

## ۷. منابع

- [1] ج. ب. و. کارسن، ج. شبیه‌سازی سیستم‌های پیشامد-گسسته. تهران: انتشارات علمی دانشگاه صنعتی شریف، ۱۳۸۴.
- [2] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of computational biology*, vol. 9, no. 1, pp. 67-103, 2002.
- [3] J. Banks and J. S. Carson, "Process-interaction simulation languages," *Simulation*, vol. 44, no. 5, pp. 225-234, 1985.
- [4] J. April, M. Better, F. Glover, and J. Kelly, "New advances and applications for marrying simulation and optimization," in *Simulation Conference, 2004. Proceedings of the 2004 Winter*, 2004, vol. 1: IEEE.
- [5] M. C. Fu, F. W. Glover, and J. April, "Simulation optimization: a review, new developments, and applications," in *Proceedings of the 37th conference on Winter simulation*, 2005, pp. 83-95: Winter Simulation Conference.
- [6] P. L'Ecuyer and P. W. Glynn, "Stochastic optimization by simulation: Convergence proofs for the GI/G/1 queue in steady-state," *Management Science*, vol. 40, no. 11, pp. 1562-1578, 1994.
- [7] V. C. Chen, K.-L. Tsui, R. R. Barton, and M. Meckesheimer, "A review on design, modeling and applications of computer experiments," *IIE transactions*, vol. 38, no. 4, pp. 273-291, 2006.
- [8] M. Bazaraa, H. Sherali, and C. Shetty, "Practical methods of Optimization: theory and algorithms," ed: John Wiley and Sons, 2006.
- [9] M. C. Fu, *Handbook of simulation optimization*. Springer, 2015.

- [10] S. G. Henderson and B. L. Nelson, *Handbooks in operations research and management science: simulation*. Elsevier, 2006.
- [11] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, 1995, pp. 39-43: IEEE.
- [12] S. Andradóttir, "Simulation optimization," *Handbook of simulation: Principles, methodology, advances, applications, and practice*, pp. 307-333, 1998.
- [13] X. Li, C. Gong, L. Gu, Z. Jing, H. Fang, and R. Gao, "A reliability-based optimization method using sequential surrogate model and Monte Carlo simulation," *Structural and Multidisciplinary Optimization*, pp. 1-22, 2018.
- [14] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of global optimization*, vol. 34, no. 3, pp. 441-466, 2006.
- [15] J. Park and K.-Y. Kim, "Meta-modeling using generalized regression neural network and particle swarm optimization," *Applied Soft Computing*, vol. 51, pp. 354-369, 2017.
- [16] G. Box and K. Wilson, "On the experimental attainment of optimum conditions," *Royal Statistical Society B*, vol. 13, no. 1, p. 45, 1951.
- [17] J. P. Kleijnen, "Response surface methodology for constrained simulation optimization: An overview," *Simulation Modelling Practice and Theory*, vol. 16, no. 1, pp. 50-64, 2008.
- [18] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.
- [19] G. Matheron, "Principles of geostatistics," *Economic geology*, vol. 58, no. 8, pp. 1246-1266, 1963.
- [20] J. P. Kleijnen, "Kriging metamodeling in simulation: A review," *European journal of operational research*, vol. 192, no. 3, pp. 707-716, 2009.

- [21] G. E. Box and D. W. Behnken, "Some new three level designs for the study of quantitative variables," *Technometrics*, vol. 2, no. 4, pp. 455-475, 1960.
- [22] F. A. Viana, "A tutorial on Latin hypercube design of experiments," *Quality and Reliability Engineering International*, vol. 32, no. 5, pp. 1975-1985, 2016.
- [23] Y. Lin, J. Allen, and F. Mistree, "Metamodel validation with deterministic computer experiments," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, p. 5425.
- [24] A. F. Daughety and M. A. Turnquist, "Simulation optimization using response surfaces based on spline approximations," *ACM SIGSIM Simulation Digest*, vol. 10, no. 1-2, pp. 73-82, 1978.
- [25] P. W. Glynn, "Optimization of stochastic systems via simulation," in *Proceedings of the 21st conference on Winter simulation*, 1989, pp. 90-105: ACM.
- [26] G. Tompkins and F. Azadivar, "Genetic algorithms in optimizing simulated systems," in *Proceedings of the 27th conference on Winter simulation*, 1995, pp. 757-762: IEEE Computer Society.
- [27] R. J. Paul and T. S. Chaney, "Simulation optimisation using a genetic algorithm," *Simulation Practice and Theory*, vol. 6, no. 6, pp. 601-611, 1998.
- [28] S. Bashyam and M. C. Fu, "Optimization of (s, S) inventory systems with random lead times and a service level constraint," *Management Science*, vol. 44, no. 12-part-2, pp. S243-S256, 1998.
- [29] L. Wang, "A hybrid genetic algorithm–neural network strategy for simulation optimization," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1329-1343, 2005.
- [30] M. A. Ahmed and T. M. Alkhamis, "Simulation optimization for an emergency department healthcare unit in Kuwait," *European journal of operational research*, vol. 198, no. 3, pp. 936-942, 2009.

- [31] R. R. Barton and M. Meckesheimer, "Metamodel-based simulation optimization," *Handbooks in operations research and management science*, vol. 13, pp. 535-574, 2006.
- [32] K.-H. Chang, L. J. Hong, and H. Wan, "Stochastic trust region gradient-free method (strong): a new response-surface-based algorithm in simulation optimization," in *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, 2007, pp. 346-354: IEEE Press.
- [33] H.-G. Beyer and B. Sendhoff, "Robust optimization—a comprehensive survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33-34, pp. 3190-3218, 2007.
- [34] J. P. Kleijnen, W. Van Beers, and I. Van Nieuwenhuysse, "Constrained optimization in expensive simulation: Novel approach," *European journal of operational research*, vol. 202, no. 1, pp. 164-174, 2010.
- [35] B. Can and C. Heavey, "A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models," *Computers & Operations Research*, vol. 39, no. 2, pp. 424-436, 2012.
- [36] F. Altiparmak, B. Dengiz, and A. A. Bulgak, "Optimization of buffer sizes in assembly systems using intelligent techniques," in *Simulation Conference, 2002. Proceedings of the Winter*, 2002, vol. 2, pp. 1157-1162: IEEE.
- [37] L. Wang, S. Li, F. Tian, and X. Fu, "A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2119-2125, 2004.
- [38] B. Dengiz, C. Alabas-Uslu, and O. Dengiz, "Optimization of manufacturing systems using a neural network metamodel with a new training approach," *Journal of the Operational Research Society*, vol. 60, no. 9, pp. 1191-1197, 2009.
- [39] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219-241, 2010.



- [40] A. C. Keys and L. P. Rees, "A sequential-design metamodeling strategy for simulation optimization," *Computers & Operations Research*, vol. 31, no. 11, pp. 1911-1932, 2004.
- [41] J. Villemonteix, E. Vazquez, and E. Walter, "An informational approach to the global optimization of expensive-to-evaluate functions," *Journal of Global Optimization*, vol. 44, no. 4, p. 509, 2009.
- [42] S. Moghaddam and H. Mahlooji, "A new metamodel-based method for solving semi-expensive simulation optimization problems," *Communications in Statistics-Simulation and Computation*, vol. 46, no. 6, pp. 4795-4811, 2017.
- [43] L. Willmes, T. Back, Y. Jin, and B. Sendhoff, "Comparing neural networks and kriging for fitness approximation in evolutionary optimization," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, 2003, vol. 1, pp. 663-670: IEEE.
- [44] R. Yuan and B. Guangchen, "Comparison of neural network and Kriging method for creating simulation-optimization metamodels," in *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*, 2009, pp. 815-821: IEEE.
- [45] E. Rigoni and A. Lovison, "Automatic sizing of neural networks for function approximation," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 2005-2010: IEEE.
- [46] M. Meckesheimer, A. J. Booker, R. R. Barton, and T. W. Simpson, "Computationally inexpensive metamodel assessment strategies," *AIAA journal*, vol. 40, no. 10, pp. 2053-2060, 2002.
- [47] W. Q. Meeker and L. A. Escobar, "Teaching about approximate confidence regions based on maximum likelihood estimation," *The American Statistician*, vol. 49, no. 1, pp. 48-53, 1995.
- [48] O. Sahin and B. Akay, "Comparisons of metaheuristic algorithms and fitness functions on software test data generation," *Applied Soft Computing*, vol. 49, pp. 1202-1214, 2016.
- [49] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

- [50] A. O. Griewank, "Generalized descent for global optimization," *Journal of optimization theory and applications*, vol. 34, no. 1, pp. 11-39, 1981.
- [51] L. A. Rastrigin, "Systems of extremal control," *Mir, Moscow*, 1974.
- [52] J. M. Dieterich and B. Hartke, "Empirical review of standard benchmark functions using evolutionary global optimization," *arXiv preprint arXiv:1207.4318*, 2012.
- [53] R. V. Hogg and J. Ledolter, *Engineering statistics*. Macmillan Pub Co, 1987.

## Abstract

Although a considerable number of problems whose analysis depends on a set of complex mathematical relations exist in the literature due to recent developments in the field of decision making, still very simplified and unrealistic assumptions are involved in many. Simulation is one of the most powerful tools to deal with this kind of problems and enjoys being free of any restricting assumptions which may generally be considered in a stochastic system. In addition, simulation optimization techniques are categorized into two broad classes of model-based and metamodel-based methods. In the first class, simulation and optimization component interact with each other causing an increase in simulation times and costs. To cope with this problem, a third component defined as metamodel that estimates the relationships between the inputs and outputs of the system being simulated, comes to the picture in the second class problems. Besides, optimization of semi-expensive simulation optimization problems needs a numerous simulation run in model-based methods. However, as the validation cost increases at a rapid rate in each iteration of the metamodel-based methods, a new metamodel-based method for solving semi-expensive simulation optimization problems has been introduced in the literature which consists of two phases and solve the problem in a less computational time. In the first phase, as a model-based algorithm, the simulation output is used directly in the optimization stage. In the second phase, the simulation model is changed with a validated metamodel. In this thesis, an artificial neural network is employed as the metamodel and its performance is compared with the original algorithm that uses a Kriging metamodel.

**Keywords:** Semi expensive simulation problems, Simulation optimization, Metamodel-based algorithm, Artificial neural network

*Approval Sheet*

*In the name of GOD*

*Sharif University of Technology*

*(Department of Industrial Engineering)*

*M.Sc. Thesis*

**A thesis submitted in partial fulfillment of the requirement for the M.Sc. degree**

***Title:*** An artificial neural network meta-model for solving semi expensive simulation optimization problems

***Author:*** Mohammad Behbahani

***Referees Committee:***

**- Supervisor: Dr.Seyed Taghi Akhavan Niaki**

**signature.....**

**- Dr.Hashem Mahlooji**

**signature.....**

**- Dr.Hamidreza Eskandari**

**signature.....**

**Date.....**



**Sharif University of Technology**  
**(Department Of Industrial Engineering)**

**A thesis submitted in partial fulfillment of the requirement for the M.Sc.  
degree**

**An artificial neural network meta-model for solving semi  
expensive simulation optimization problems**

**By:**

**Mohammad Behbahani**

**Supervisor:**

**Dr.Seyed Taghi Akhavan Niaki**

September 2018