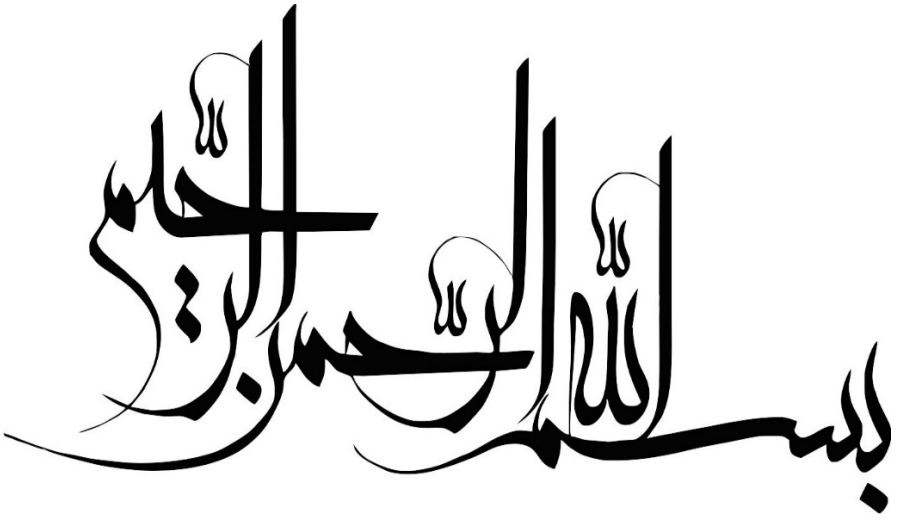
بسم الله الرحمن الرحيم

الگوریتم‌های فراابتکاری در یادگیری ماشین
پیشرفت‌ها و ابزارهای نوین

Metaheuristics for Machine Learning
New Advances and Tools

منصور ادالی، باسم جربوئی، پاتریک سیاری

علی ماروسی

(استادیار، گروه مهندسی کامپیوتر، دانشگاه تربت حیدریه)

۱۴۰۴

عنوان و نام پدیدآور	:	**الگوریتم‌های فراابتکاری در یادگیری ماشین: پیشرفت‌ها و ابزارهای نوین/ نویسندگان [صحیح: ویراستاران] منصور ادالی، باسم جربوئی، پاتریک سیاری؛ مترجم علی ماروسی.**
مشخصات نشر	:	**تربت حیدریه: دانشگاه تربت حیدریه، انتشارات، ۱٤۰٤ سال.**
مشخصات ظاهری	:	**[۲۹]، ٤۰۳ ص.: مصور، جدول، نمودار.**
شابک	:	600-8335-48-1978-
وضعیت فهرست نویسی	:	**فیپا**
یادداشت		**عنوان اصلی: Metaheuristics for machine learning: new advances and tools, 2023.**
موضوع	:	**فراگیری ماشین** **Machine learning** **الگوریتم‌های فراابتکاری** **Metaheuristics** **هوش مصنوعی** **Artificial intelligence**
شناسه افزوده	:	ادالی، منصور، ویراستار
شناسه افزوده	:	Eddaly, Mansour
شناسه افزوده	:	جربوئی، باسم، ویراستار
شناسه افزوده	:	Jarboui, Bassem
شناسه افزوده	:	سیاری، پاتریک، ۱۹۵۲- م، ویراستار
شناسه افزوده	:	Siarry, Patrick, 1952-
شناسه افزوده	:	ماروسی، علی، ۱۳۶۰-، مترجم
شناسه افزوده	:	دانشگاه تربت‌حیدریه. انتشارات
رده بندی کنگره	:	**۳۲۵/٥Q**
رده بندی دیویی	:	**۰۰٦/۳۱**
شماره کتابشناسی ملی	:	**۱۰۲۸۲۰۸٤**
اطلاعات رکورد کتابشناسی	:	**فیپا**

عنوان کتاب: الگوریتم‌های فراابتکاری در یادگیری ماشین پیشرفت‌ها و ابزارهای نوین
نویسنده: منصور ادالی، باسم جربوئی، پاتریک سیاری
مترجم/مترجمان: علی ماروسی
چاپ اول
بها:
نشانی ناشر: تربت حیدریه، کیلومتر هفت جاده مشهد، دانشگاه دولتی تربت حیدریه

پیشگفتار

الگوریتمهای فراابتکاری به‌عنوان ابزاری کارآمد برای حل مسائل پیچیده بهینه‌سازی شناخته شده‌اند. ایده اصلی آن‌ها حفظ تعادل بین تنوع بخشی و تمرکز (تشدید) گرایی است تا راه‌حل‌های باکیفیتی را در زمان معقولی ارائه دهند. تحقیق و توسعه در زمینه استفاده از الگوریتم‌های فراابتکاری برای بهبود روش‌های یادگیری ماشین به موضوعی رایج تبدیل شده است. برخی از کاربردهای موفق آن‌ها در مسائل تحت نظارت (طبقه‌بندی و رگرسیون) و غیر تحت نظارت (خوشه‌بندی و کشف قوانین) به کار می‌روند. علاوه بر این، تولید خودکار برنامه‌ها از طریق الگوریتمهای فراابتکاری، مانند محاسبات تکاملی و هوش جمعی، محبوبیت زیادی پیدا کرده است. این موفقیت عمدتاً توسط پارادایم برنامه‌نویسی ژنتیکی که توسط کوزا در سال ۱۹۹۲ توسعه یافت، ترویج شده است.

در این کتاب، روش‌های مختلف استفاده از الگوریتمهای فراابتکاری در روش‌های یادگیری ماشین از جنبه‌های نظری و عملی مورد بررسی قرار می‌گیرد. این کتاب آخرین نوآوری‌ها و کاربردهای ادغام الگوریتمهای فراابتکاری در روش‌های یادگیری ماشین را مرور می‌کند و به برنامه‌نویسی فراابتکاری، فرایادگیری و غیره می‌پردازد. علاوه بر این، برخی از مثال‌ها از کاربردهای واقعی مانند خوشه‌بندی، داده‌های کلان، نظارت بر سلامت ماشین، شناسایی اهداف سونار زیرآبی، بانکداری و غیره ارائه می‌شود.

ساختار کتاب

این کتاب به دو بخش اصلی تقسیم می‌شود: بخش اول "الگوریتمهای فراابتکاری برای یادگیری ماشین: نظریه و مرورها" شامل سه فصل است و به جنبه‌های نظری می‌پردازد، درحالی‌که بخش دوم "الگوریتمهای فراابتکاری برای یادگیری ماشین: کاربردها" شامل پنج فصل است و برخی از کاربردهای دنیای واقعی را بررسی می‌کند.

فصل ۱ با عنوان "از الگوریتمهای فراابتکاری تا برنامه‌نویسی خودکار"، مروری بر مهم‌ترین الگوریتمهای فراابتکاری ارائه داده و توضیح می‌دهد چگونه می‌توان آن‌ها را برای تکامل برنامه‌ها تطبیق داد. همچنین مقایسه‌ای دقیق بین الگوریتمهای فراابتکاری برنامه‌نویسی خودکار و روش‌های سنتی یادگیری ماشین انجام می‌دهد تا مزایای استفاده از هر یک را توضیح دهد.

فصل ۲ با عنوان "الگوریتم‌های بدون خوشه‌بندی مبتنی بر الگوریتمهای فراابتکاری: یک مرور جامع"، رویکردهای فراابتکاری برای حل مسأله بدون خوشه‌بندی ارائه می‌دهد. این مرور بر روش‌های بهینه‌سازی زیربنایی و جستجوی اصلی آن‌ها تمرکز دارد و همچنین بحثی خاص در مورد رویکردهای تک‌هدفه در مقابل چندهدفه ارائه می‌دهد.

فصل ۳ با عنوان "دیدگاه فراابتکاری در سیستم‌های یادگیرنده طبقه‌بندی‌کننده" به بررسی سیستم‌های یادگیرنده طبقه‌بندی‌کننده می‌پردازد که یک خانواده از سیستم‌های یادگیری مبتنی بر قوانین هستند. همچنین شباهت‌ها و تفاوت‌های LCSها و روش‌های یادگیری ماشین مربوط به برنامه‌نویسی ژنتیکی، درخت‌های تصمیم، ترکیب‌های خبره، بگینگ و بوستینگ مورد بحث قرار می‌گیرد.

فصل ٤ با عنوان "رویکرد مبتنی بر الگوریتم فراابتکاری برای یادگیری ماشین جهت خوشه‌بندی مشتریان" یک رویکرد تکاملی خوشه‌بندی را به‌عنوان یک مکانیزم استخراج قوانین پیشنهاد می‌دهد که به تصمیم‌گیرندگان کمک می‌کند تا ویژگی‌های مهم مشتری را شناسایی کرده و آن‌ها را در خوشه‌ها طبقه‌بندی کنند. درنتیجه، از الگوریتم ژنتیک در ترکیب با الگوریتم‌های یادگیری ماشین بدون نظارت (الگوریتم‌های K-means) برای حل مسائل خوشه‌بندی داده‌ها استفاده می‌شود.

فصل ٥ با عنوان "تکامل طبقه‌بندی‌کننده‌های مبتنی بر یادگیری ماشین از طریق رویکرد فراابتکاری برای شناسایی اهداف سونار زیرآبی" عملکرد ده الگوریتم فراابتکاری را برای استفاده در شبکه‌های ماشین بردار پشتیبان (SVM) بررسی می‌کند تا یک طبقه‌بندی‌کننده هدف‌های سونار زیرآبی دقیق و قابل اعتماد داشته باشد.

فصل ٦ با عنوان "حل مسأله کوله‌پشتی درجه‌دو با استفاده از یک رویکرد جستجوی تطبیقی تصادفی حریصانه (GRASP) برای حل مساله کوله‌پشتی درجه‌دو (QKP) ارائه می‌دهد. علاوه بر این، یک جستجوی محلی جدید توسعه داده شده است که توانسته است همسایگی بزرگ‌تری از راه‌حل‌ها را جستجو کند و در عین حال همان هزینه محاسباتی پیچیدگی زمانی را که معمولاً استفاده می‌شود، حفظ کند.

فصل ٧ با عنوان "الگوریتم در مقابل مدیریت پردازش برای مقیاس‌بندی برنامه‌نویسی ژنتیکی به داده‌های کلان" یک طبقه‌بندی برای دسته‌بندی راه‌حل‌های مسائل هنگام آموزش الگوریتم‌های تکاملی/برنامه‌نویسی ژنتیکی بر روی مجموعه‌های داده بزرگ در سه دسته پردازش، پردازش الگوریتم و مدیریت داده‌ها پیشنهاد می‌دهد. دو رویکرد سپس ارائه و بررسی می‌شود: اولی از دسته مدیریت پردازش است که برنامه‌نویسی ژنتیکی را روی Spark موازی می‌کند، دومی از دسته پردازش الگوریتم است که برنامه‌نویسی ژنتیکی را با یادگیری فعال گسترش می‌دهد و از نمونه‌برداری پویا و تطبیقی استفاده می‌کند.

فصل ۸ "مسئله تخصیص پویا پارکینگ" یک مدل فرمول‌بندی MIP با تقسیم زمان و مجموعه‌ای از نقاط تصمیم برای مدیریت جنبه پویا این مسئله پیشنهاد می‌دهد. برای حل این مسأله، یک رویکرد ترکیبی با استفاده از الگوریتم تخصیص مونکر، جستجوی محلی و الگوریتم توزیع تخمینی (EDA) همراه با یادگیری تقویتی ارائه شده است. نتایج به‌دست‌آمده نشان می‌دهند که این رویکرد با اثر یادگیری، کارآمد است.

منصور ادالی

باسم جربوئی

پاتریک سیاری

معرفی همکاران

عبدالصمد آیت الکدی، دانشگاه پلی‌تکنیک اُتو-د-فرانس، LAMIH، CNRS، والنسین، فرانسه
INSA اُتو-د-فرانس، والنسین، فرانسه
سانا بن حمیده، LAMSADE، دانشگاه پاریس دوفین، PSL Research University، پاریس، فرانسه
کلاریس دهنین، دانشگاه لیل، CNRS، سنترال لیل، UMR 9189 CRIStAL، لیل، فرانسه
منصور ادالی، دپارتمان سیستم‌های اطلاعات مدیریت و مدیریت تولید، دانشکده اقتصاد و تجارت، دانشگاه قاسم، بریده، پادشاهی عربستان سعودی
سوهیر الوچ، دانشکده اقتصاد و تجارت، دانشگاه قاسم، بریده، پادشاهی عربستان سعودی
یورگ هانه، دانشگاه آوگسبورگ، آوگسبورگ، آلمان
مایکل هایدر، دانشگاه آوگسبورگ، آوگسبورگ، آلمان
حمیده، مؤسسه عالی مطالعات فناوری بی‌زرته، دپارتمان عبدالرحن، تونس
ژولی ژاک، دانشگاه لیل، CNRS، سنترال لیل، UMR 9189 CRIStAL، لیل، فرانسه
باسم جربوی، کالج‌های عالی فناوری، ابوظبی، امارات متحده عربی
حامد جاودانفر، دپارتمان مهندسی الکترونیک دریایی و مهندسی ارتباطات، دانشگاه علوم دریایی امام خمینی، نوشهر، ایران
آدان خوزه-گارثیا، دانشگاه لیل، CNRS، لیل، فرانسه
راکا جووانوویچ، مؤسسه تحقیقاتی محیط زیست و انرژی قطر (QEERI)، دانشگاه حماد بن خلیفه، دوحه، قطر
محمد کاظمی‌راد، دپارتمان مهندسی الکترونیک دریایی و مهندسی ارتباطات، دانشگاه علوم دریایی امام خمینی، نوشهر، ایران
محمد کیشه، دپارتمان مهندسی الکترونیک دریایی و مهندسی ارتباطات، دانشگاه علوم دریایی امام خمینی، نوشهر، ایران

پی. ز. لاپاس، دپارتمان آمار و ریاضیات مالی-اکچواری، دانشگاه ایجه، ساموس، یونان
EXUS AI Labs، EXUS، آتن، یونان
حسن محمدی، دپارتمان مهندسی الکترونیک دریایی و مهندسی ارتباطات، دانشگاه علوم دریایی امام خمینی، نوشهر، ایران
دیوید پاتزل، دانشگاه آوگسبورگ، آوگسبورگ، آلمان
مصطفی رطلی، دانشگاه پلی‌تکنیک اُتو-د-فرانس، LAMIH، CNRS، والنسین، فرانسه
پاتریک سیاری، LISSI، دانشگاه پاریس اِست کریتی، ویتری-سور-سن، فرانسه
وینسنت سوبانسکی، دانشگاه لیل، Inserm، CHU لیل، مؤسسه دانشگاهی فرانسه - U1286 (IUF)
INFINITE – مؤسسه تحقیقاتی انتقالی در التهاب، لیل، فرانسه
هلنا اشتگر، دانشگاه آوگسبورگ، آوگسبورگ، آلمان
استفان فاس، موسسه سیستم‌های اطلاعاتی، دانشگاه هامبورگ، هامبورگ، آلمان
س. ز. زانتوپولوس، دپارتمان آمار و ریاضیات مالی-اکچواری، دانشگاه ایجه، ساموس، یونان
آ. ن. یاناکوپولوس، دپارتمان آمار و آزمایشگاه مدل‌سازی تصادفی و کاربردها، دانشگاه اقتصاد و تجارت آتن، آتن، یونان

پیشگفتار مترجم

کتاب حاضر با عنوان *Metaheuristics for Machine Learning: New Advances and Tools* به بررسی پیشرفت‌های نوین و ابزارهای به‌کاررفته در زمینه بهینه‌سازی‌های الگوریتمهای فراابتکاری و کاربرد آن‌ها در یادگیری ماشین می‌پردازد. الگوریتم‌های فراابتکاری به‌عنوان ابزاری کارآمد برای حل مسائل پیچیده بهینه‌سازی شناخته شده‌اند که در آن‌ها تلاش می‌شود تا تعادل میان گسترش جستجو و تمرکز بر روی نواحی خاص از فضای حل حفظ شود تا بتوان راه‌حل‌های باکیفیت را در مدت‌زمان معقولی ارائه داد. این روش‌ها به‌ویژه در سال‌های اخیر توجه بسیاری از محققان را به خود جلب کرده‌اند، چرا که استفاده از الگوریتمهای فراابتکاری در بهبود روش‌های یادگیری ماشین به‌سرعت در حال رشد و توسعه است.

در این کتاب، سعی شده است تا جدیدترین نوآوری‌ها و کاربردهای الگوریتمهای فراابتکاری در یادگیری ماشین از جنبه‌های نظری و عملی بررسی شود. در فصول مختلف، علاوه بر توضیحات نظری در مورد روش‌های فراابتکاری و نحوه به‌کارگیری آن‌ها، به کاربردهای واقعی این روش‌ها در مسائل مختلفی چون خوشه‌بندی، تحلیل داده‌های بزرگ، پایش سلامت ماشین، شناسایی اهداف سونار زیرآبی و مسائل بانکی پرداخته شده است. این کتاب شامل دو بخش اصلی است: بخش اول با عنوان *Metaheuristics for Machine Learning: Theory and Reviews* به جنبه‌های نظری و مرورهای علمی اختصاص دارد و بخش دوم تحت عنوان *Metaheuristics for Machine Learning: Applications* به بررسی کاربردهای عملی و واقعی الگوریتمهای فراابتکاری در زمینه‌های مختلف می‌پردازد.

از ویژگی‌های برجسته این کتاب، مقایسه روش‌های سنتی یادگیری ماشین با الگوریتمهای فراابتکاری و همچنین کاربردهای نوآورانه آن در زمینه‌های مختلف است. ازطرفی در فصل‌های مختلف کتاب، به‌طور ویژه به موضوعاتی مانند برنامه‌نویسی خودکار با استفاده از الگوریتمهای

فراابتکاری، خوشه‌بندی و تقسیم‌بندی مشتریان، شناسایی اهداف در سونارهای زیرآبی و مسائل پیچیده داده‌های بزرگ پرداخته شده است.

امید است این کتاب به‌عنوان یک مرجع کامل و جامع، برای پژوهشگران و علاقه‌مندان به یادگیری ماشین و الگوریتم‌های بهینه‌سازی فراابتکاری، ابزاری مفید در جهت ارتقاء دانش و بهبود کاربردهای عملی این روش‌ها در مسائل واقعی باشد.

فهرست

بخش اول ۱

الگوریتم‌های فراابتکاری برای یادگیری ماشین: نظریه ها و دیدگاه‌ها ۱

فصل اول ۳

از الگوریتم‌های فراابتکاری تا برنامه‌نویسی خودکار ۳

چکیده ۳

۱-۱- مقدمه ۴

۱-۲- الگوریتمهای فراابتکاری ۷

۱-۲-۱- الگوریتم‌های فراابتکاری مبتنی بر راه‌حل ۷

۱-۲-۱-۱- تبرید شبیه‌سازی‌شده ۸

۱-۲-۱-۲- جستجوی ممنوعه ۹

۱-۲-۱-۳- الگوریتم تپه‌نوردی ۱۰

۱-۲-۱-۴- جستجوی همسایگی متغیر ۱۰

۱-۲-۱-۵ جستجوی محلی تکرارشونده ۱۲

۱-۲-۲- الگوریتم‌های فراابتکاری مبتنی بر جمعیت ۱۳

۱-۲-۲-۱ محاسبات تکاملی ۱۳

الگوریتم ژنتیک ۱۴

استراتژی‌های تکاملی ۱۵

تکامل دیفرانسیلی ۱۵

۱-۲-۲-۲- هوش ازدحامی ۱۶

بهینه‌سازی کلونی مورچه‌ها ۱۶

بهینه‌سازی ازدحام ذرات ۱۷

الگوریتم کلونی زنبورهای مصنوعی..... ۱۸
سیستم‌های ایمنی مصنوعی ۲۰
۱-۳- برنامه‌نویسی خودکار..... ۲۱
۱-۳-۱- نمایش برنامه ۲۲
۱-۳-۱-۱- نمایش‌های مبتنی بر ساختار درختی..... ۲۲
۱-۳-۱-۲- نمایش‌های خطی..... ۲۳
۱-۳-۱-۳- سایر نمایش‌ها..... ۲۵
۱-۳-۲- برنامه‌نویسی ژنتیکی ۲۶
۱-۳-۳- برنامه‌نویسی ایمنی..... ۲۸
۱-۳-۴- برنامه‌نویسی بیان ژن ۳۰
۱-۳-۵- برنامه‌نویسی ازدحام ذرات ۳۱
۱-۳-۶- برنامه‌نویسی کلونی زنبور مصنوعی ۳۲
۱-۳-۷- برنامه‌نویسی کلونی مورچه‌ها ۳۳
۱-۳-۸- سایر الگوریتم‌های برنامه‌نویسی خودکار ۳۴
۱-۳-۸-۱- برنامه‌نویسی باکتریایی..... ۳۵
۱-۳-۸-۳- تکامل دیفرانسیل مبتنی بر درخت..... ۳۶
۱-۳-۸-۴- برنامه‌نویسی تخمین توزیع ۳۷
۱-۳-۸-۵- الگوریتم آتش‌بازی گرامری..... ۳۸
۱-۳-۸-۶- برنامه‌نویسی جمعیت ماهی مصنوعی..... ۳۸
۱-۳-۸-۷- برنامه‌نویسی همسایگی متغیر ۳۹
۱-۴- بحث و چالش‌ها ۴۰
۱-۵- نتیجه‌گیری ۴۴

منابع ۴۴
فصل دوم ٦١
مروری بر الگوریتم‌های خوشه‌بندی دوگانه مبتنی بر الگوریتمهای فراابتکاری ۶۱
چکیده ۶۱
۲-۱- مقدمه ۶۲
۲-۲- مقدمات پایه ۶۵
۲-۲-۱ تعریف خوشه‌بندی دوگانه ۶۵
۲-۲-۲- روش‌های کلاسیک خوشه‌بندی دوگانه ۶۷
۲-۲-۲-۱- ترکیب خوشه‌بندی سطری و ستونی تکراری ۶۸
۲-۲-۲-۲- تقسیم و تسخیر ۶۸
۲-۲-۲-۳- جستجوی تکراری حریصانه ۶۹
۲-۲-۲-۴- شمارش فراگیر خوشه‌های‌دوگانه ۶۹
۲-۲-۲-۵- شناسایی پارامترهای توزیع ۷۰
۲-۲-۳- خوشه‌بندی دوگانه به‌عنوان یک مسئله بهینه‌سازی ۷۰
۲-۳- جزء اصلی الگوریتم‌های فراابتکاری برای خوشه‌بندی دوسطحی ۷۱
۲-۳-۱- کدگذاری خوشه‌بندی دوسطحی ۷۲
۲-۳-۱-۱- کدگذاری دودویی خوشه‌بندی دوسطحی (BBE) ۷۲
۲-۳-۱-۲- کدگذاری عدد صحیح خوشه‌بندی دوسطحی ۷۴
۲-۳-۲- عملگرهای تغییر ۷۴
۲-۳-۳- عملگرهای ترکیب ۷۵
۲-۳-۳-۱- عملگر ترکیب تک نقطه‌ای ۷۵
۲-۳-۳-۲- عملگر ترکیب خوشه‌بندی دوسطحی ۷۵

۴-۳-۲- عملگرهای جهش ۷۵
۲-۳-۴-۱- عملگر جهش تصادفی ۷۶
۲-۳-۴-۲- عملگر جهش مبتنی بر CC ۷۶
۵-۳-۲- توابع هدف ۷۶
۲-۳-۵-۱- اندازه خوشه دوسطحی ۷۷
۲-۳-۵-۲- واریانس خوشه دوسطحی (VAR) ۷۷
۲-۳-۵-۳- میانگین مربعات باقیمانده (MSR) ۷۸
۲-۳-۵-۴- میانگین مربعات باقیمانده مقیاس‌بندی شده ۷۸
۲-۳-۵-۵- تابع همبستگی میانگین (ACF) ۷۸
۲-۳-۵-۶- مقدار همبستگی میانگین (ACV) ۷۹
۲-۳-۵-۷- خطای مجازی ۷۹
۸-۵-۳-۲- تابع ضریب تغییرات ۸۰
۴-۲- خوشه‌بندی دوسطحی تک‌هدفه ۸۰
۱-۴-۲- شبیه‌سازی تبرید ۸۲
۲-۴-۲- الگوریتم‌های ژنتیک ۸۲
۳-۴-۲- جستجوی پراکنده ۸۷
۴-۴-۲- جستجوی فاخته ۸۹
۵-۴-۲- بهینه‌سازی جمعی ذرات ۸۹
۶-۴-۲- رویکردهای فراابتکاری ترکیبی ۹۰
۷-۴-۲- خلاصه ۹۲
۵-۲- خوشه‌بندی‌دوگانه چندهدفه ۹۳
۱-۵-۲- الگوریتم‌های تکاملی چندهدفه مبتنی بر NSGA-II ۹۴

۲-۵-۲- الگوریتم‌های تکاملی چندهدفه مبتنی بر SPEA2 و IBEA ۹۸
۳-۵-۲- بهینه‌سازی چندهدفه به روش جمعیت ذرات ۱۰۰
۴-۵-۲- سایر رویکردهای چندهدفه ۱۰۱
۵-۵-۲- خلاصه ۱۰۳
۶-۲- بحث و جهت‌گیری‌های آینده ۱۰۴
۱-۶-۲- مسیرهای آینده ۱۰۴
۷-۲- نتیجه‌گیری ۱۰۶
منابع ۱۰۷
فصل سوم ۱۱۵
ساخت سیستم‌های تصمیم‌گیری هوشمند با الگوریتم‌های فراابتکاری ۱۱۵
چکیده ۱۱۵
۱-۳- انگیزه و ساختار ۱۱۶
۲-۳- سیستم‌های طبقه‌بندی یادگیری چیست؟ ۱۱۷
۱-۲-۳- وظایف یادگیری ۱۱۹
۲-۲-۳- مدل‌های LCS ۱۲۰
۳-۲-۳- ساختار الگوریتمی کلی LCS ۱۲۲
۴-۲-۳- LCSها در ادبیات فراابتکاری ۱۲۵
۳-۳- سیستم‌های یادگیری ماشین مشابه LCSها ۱۲۶
۱-۳-۳- درخت‌های تصمیم‌گیری ۱۲۶
۲-۳-۳- ترکیب مدل‌ها ۱۲۸
۳-۳-۳- بگینگ و بوستینگ ۱۳۰

۴-۳-۳- برنامه‌نویسی ژنتیک ۱۳۱
۴-۳- نقش الگوریتمهای فراابتکاری در LCSها ۱۳۲
۴-۴-۳- چهار نوعLCS ۱۳۳
۴-۴-۳- نمایندگی راه‌حل‌های فراابتکاری ۱۳۶
۴-۴-۳- عملگرهای فراابتکاری ۱۳۷
۴-۴-۳- توابع تناسب معمول ۱۴۱
۵-۳- سایر سیستم‌های یادگیری ماشین مبتنی بر قوانین با تمرکز بر الگوریتمهای فراابتکاری ۱۴۳
۱-۵-۳- رویکردهای مبتنی بر الگوریتم‌های تکاملی ۱۴۴
۲-۵-۳- رویکردهای مبتنی بر سیستم مورچه‌ها ۱۴۵
۳-۵-۳- رویکردهای مبتنی بر الگوریتمهای فراابتکاری دیگر یا ترکیبی ۱۴۶
۴-۵-۳- سیستم‌های ایمنی مصنوعی ۱۴۸
۶-۳- پژوهش‌های آتی ۱۴۹
۷-۳- نتیجه‌گیری ۱۵۰
منابع ۱۵۲
بخش دوم ۱۵۹
الگوریتم‌های فراابتکاری برای یادگیری ماشین: کاربردها ۱۵۹
فصل چهارم ۱۶۱
رویکرد یادگیری ماشین مبتنی بر الگوریتم‌های فراابتکاری برای طبقه‌بندی مشتریان ۱۶۱
چکیده ۱۶۱
۴-۱- مقدمه ۱۶۲

۴-۲- روش پیشنهادی ۱۷۳
۴-۲-۱- انتخاب ویژگی ۱۷۳
۴-۲-۲- خوشه‌بندی ۱۷۸
۴-۲-۳- ساختار کلی طرح ترکیبی پیشنهادی ۱۸۰
۴-۲-۴- طبقه‌بندی مبتنی بر خوشه‌بندی ۱۸۳
۴-۳- آزمون‌های محاسباتی و نتایج ۱۸۵
۴-۳-۱- تنظیمات آزمایش ۱۸۵
۴-۳-۲- مجموعه داده‌های بررسی‌شده ۱۸۵
۴-۳-۳- تنظیمات پارامترها ۱۸۸
۴-۳-۴- نتایج و بحث ۱۸۸
۴-۳-۴-۱- آزمایش ۱ ..... ۱۸۸
۴-۴-۳-۴- آزمایش ۲ ..... ۱۹۰
نتایج خوشه‌بندی ۱۹۰
۴-۴- نتیجه‌گیری و زمینه‌های پژوهشی آتی ۱۹۵
منابع ۱۹۶
فصل پنجم ۲۰۵
توسعه‌ی طبقه‌بندی‌کننده‌های مبتنی بر یادگیری ماشین با رویکرد فراابتکاری ۲۰۵
چکیده ۲۰۵
۵-۱- مقدمه ۲۰۶
۵-۲- نظریه SVM ۲۰۹
۵-۳- تعریف مسئله ۲۱۲

۴-۵- آزمایش‌ها و نتایج ۲۱۵
۱-۴-۵- مجموعه داده آزمایشگاهی ۲۱۷
۵-۴-۱-۱- مراحل حذف نویز و بازتاب‌ها ۲۱۸
۵-۴-۲- مجموعه دادهٔ عمومی ۲۰۱۵ (CDS2015) ۲۲۱
۳-۴-۴- مجموعه دادهٔ تجربی واقعی زیرآبی ۲۲۱
۵-۴-۴- نتایج تجربی و بحث ۲۲۳
۵-۵- نتیجه‌گیری ۲۲۵
منابع ۲۲۶
فصل ششم ۲۳۳
حل مسئله کوله‌پشتی درجه دوم با استفاده از روش GRASP ۲۳۳
چکیده ۲۳۳
۶-۱- مقدمه ۲۳۴
۶-۲- مسئله کوله‌پشتی درجه دوم ۲۳۷
۶-۳- الگوریتم حریصانه ۲۳۷
۶-۴- روش GRASP ۲۳۹
۶-۴-۱- تصادفی‌سازی ۲۳۹
۲-۴-۶- جستجوی محلی با جایگزینی راه‌حل ۲۴۰
۳-۴-۶- جستجوی محلی چند تعویضه ۲۴۲
۴-۴-۶- پیاده‌سازی ۲۴۶
۶-۵- تولید نمونه‌ها ۲۵۱
۱-۵-۶ وزن ۲۵۱

5.2سود کالا ۲۵۱
۳-۵-۶- سود زوج‌ها ۲۵۲
۶-۶- نتایج محاسباتی ۲۵۳
۱-۶-۶- مقایسه با روش‌های موجود ۲۵۳
۲-۶-۶- مقایسه جستجوهای محلی ۲۵۷
۳-۶-۶- ارزیابی نمونه‌های آزمایشی ۲۵۸
۷-۶- نتیجه‌گیری ۲۶۲
منابع ۲۶۳
فصل هفتم ۲٦۷
مقیاس‌پذیری برنامه‌نویسی ژنتیکی در داده‌کاوی داده‌های کلان ۲۶۷
چکیده ۲۶۷
۱-۷- مقدمه ۲۶۸
۲-۷- مقیاس‌گذاری الگوریتم‌های ژنتیک به داده‌های بزرگ ۲۷۰
۱-۲-۷- مدیریت داده‌ها ۲۷۱
۲-۲-۷- مدیریت پردازش ۲۷۳
۳-۲-۷-مدیریت الگوریتم ۲۷۴
۴-۲-۷- پردازش الگوریتم ۲۷۵
۳-۷-مقیاس‌بندی الگوریتم‌های تکاملی (GP) از طریق مدیریت پردازش: موازی‌سازی افقی ۲۷۶
۱-۳-۷- اسپارک در مقابل نگاشت کاهش ۲۷۶
۲-۳-۷- موازی‌سازی الگوریتم‌های تکاملی (GP) در محیط‌های توزیع‌شده ۲۷۸

۳-۳-۷- مثال پیاده‌سازی ۲۷۹
۴-۷- مقیاس‌بندی GP از طریق پردازش الگوریتم: پارادایم یادگیری..... ۲۸۰
۱-۴-۷ یادگیری فعال با نمونه‌گیری پویا تک سطحی/چند سطحی..... ۲۸۳
۷-۴-۱-۱- مثال پیاده‌سازی با DEAP.....۲۸۶
۲-۴-۷- یادگیری فعال با نمونه‌برداری تطبیقی ۲۸۷
۵-۷- ترکیب موازی‌سازی افقی و یادگیری فعال..... ۲۸۸
۱-۵-۷- نمونه‌گیری پویا در GP توزیع‌شده ۲۸۸
۲-۵-۷- مثال پیاده‌سازی نمونه‌گیری پویا با برنامه‌نویسی ژنتیک موازی‌شده..... ۲۹۰
۶-۷- بحث..... ۲۹۰
منابع ۲۹۳
فصل هشتم ۲۹۷
مسئله تخصیص پویا برای جایگاه‌های پارکینگ..... ۲۹۷
چکیده..... ۲۹۷
۱-۸- مقدمه..... ۲۹۸
۲-۸- فرمول‌بندی برنامه‌ریزی صحیح مختلط..... ۲۹۹
۳-۸- الگوریتم‌های تخمین توزیع ۳۰۶
۱-۳-۸- مدل‌های یک‌متغیره..... ۳۰۸
۱-۳-۳-۸- یادگیری افزایشی مبتنی بر جمعیت ۳۰۹
۲-۱-۳-۸- الگوریتم تپه‌نوردی تصادفی با یادگیری از بردارهای توزیع نرمال ۳۰۹
۸-۳-۱-۳- الگوریتم توزیع حاشیه‌ای یک‌متغیره.....۳۱۰
۲-۳-۸- مدل‌های دومتغیره ۳۱۰

۳-۳-۸- مدل‌های چندمتغیره ۳۱۱
۱-۳-۳-۸- تخمین الگوریتم‌های چگالی نرمال چندمتغیره (EMNA) ۳۱۱
۲-۳-۳-۸- تخمین الگوریتم‌های شبکه گوسی ۳۱۳
۳-۳-۳-۸- الگوریتم تکاملی تخمین چگالی تکراری ۳۱۳
۴-۸- الگوریتم تخمین توزیع با یادگیری تقویتی ۳۱۳
۵-۸- نتایج محاسباتی ۳۱۷
۶-۸- نتیجه‌گیری ۳۲۴
منابع ۳۲۵

فهرست اختصارات

ABC	Artificial Bee Colony	کلونی زنبور مصنوعی
ABCP	Artificial Bee Colony Programming	برنامه‌نویسی کلونی زنبور مصنوعی
ACF	Average Correlation Function	تابع همبستگی متوسط
ACO	Ant Colony Optimization	بهینه‌سازی کلونی مورچه‌ها
ACP	Ant Colony Programming	برنامه‌نویسی کلونی مورچه‌ها
ACV	Average Correlation Value	مقدار همبستگی متوسط
AHP	Analytic Hierarchy Process	فرایند تحلیل سلسله مراتبی
AI	Artificial Intelligence	هوش مصنوعی
AIS	Artificial Immune Systems	سیستم‌های ایمنی مصنوعی
ANN	Artificial Neural Network	شبکه عصبی مصنوعی
AP	Automatic Programming	برنامه‌نویسی خودکار
BBE	Binary Bicluster Encoding	کدگذاری خوشه‌بندی‌دوگانه دودویی
BBC	Bayesian BiClustering	خوشه‌بندی‌دوگانه بیزی
BEA	Bacterial Evolution Algorithm	الگوریتم تکامل باکتریایی
BiBit	Bit-Pattern Biclustering Algorithm	الگوریتم خوشه‌بندی‌دوگانه الگوی بیتی
Bimax	Binary Inclusion-Maximal Biclustering Algorithm	الگوریتم خوشه‌بندی‌دوگانه شامل بیشینه دودویی
BSize	Bicluster Size	اندازه خوشه‌بندی‌دوگانه
BUSS	Basic Under-Sampling Selection	انتخاب نمونه‌گیری کمینه پایه
CCA	Cheng and Church's Algorithm	الگوریتم چنگ و چرچ
CFG	Context-Free Grammar	گرامر متن آزاد
CGP	Cartesian Genetic Programming	برنامه‌نویسی ژنتیکی کارتزینی
ChOA	Chimp Optimization Algorithm	الگوریتم بهینه‌سازی شامپانزه‌ها
CMA-ES	Covariance Matrix Adaptation Evolution Strategy	استراتژی تکامل تطبیق ماتریس کوواریانس
CoEA	Coevolutionary Algorithms	الگوریتم‌های هم‌تکاملی
CTWC	Coupled Two-Way Clustering	خوشه‌بندی دوسویه جفت‌شده
CVF	Coefficient of Variation Function	ضریب تغییر تابع
DCA	Direct Clustering Algorithm	الگوریتم خوشه‌بندی مستقیم
DCC	Double Conjugated Clustering	خوشه‌بندی دوبل مزدوج
DE	Differential Evolution	تکامل تفاضلی

DEAP	Distributed Evolutionary Algorithms in Python	الگوریتم‌های تکاملی توزیع‌شده در پایتون
DeBi	Differentially Expressed Biclusters	خوشه‌بندی‌های‌دوگانه بیان‌شده متفاوت
DSS	Dynamic Subset Selection	انتخاب زیرمجموعه پویا
DT	Decision Tree	درخت تصمیم
EA	Evolutionary Algorithms	الگوریتم‌های تکاملی
EC	Evolutionary Computation	محاسبات تکاملی
EDA	Estimation of Distribution Algorithm	الگوریتم تخمین توزیع
EDP	Estimation of Distribution Programming	برنامه‌نویسی تخمین توزیع
ES	Evolution Strategies	استراتژی‌های تکاملی
FABIA	Factor Analysis for Bicluster Acquisition	تحلیل عاملی برای دستیابی به خوشه‌بندی‌دوگانه
FIFO	First In First Out	اولین ورود، اولین خروج
GA	Genetic Algorithms	الگوریتم‌های ژنتیکی
GBC	Grammatical Bee Colony	کلونی زنبور گرامری
GE	Grammatical Evolution	تکامل گرامری
GEP	Gene Expression Programming	برنامه‌نویسی بیان ژن
GP	Genetic Programming	برنامه‌نویسی ژنتیکی
GPGPU	General Purpose Graphics Processing Units	واحد پردازش گرافیکی چندمنظوره
GRASP	Greedy Randomized Adaptive Search Procedure	روش جستجوی حریصانه تصادفی تطبیقی
GS	Grammatical Swarm	ازدحام گرامری
GWO	Gray Wolf Optimizer	بهینه‌ساز گرگ خاکستری
HDFS	Hadoop Distributed Files System	سیستم فایل توزیع‌شده هادوپ
HGSO	Henry Gas Solubility Optimization	بهینه‌سازی حلالیت گاز هنری
ILS	Iterated Local Search	جستجوی محلی تکراری
ITWC	Interrelated Two-Way Clustering	خوشه‌بندی دوسویه مرتبط
KF	Kalman Filter	فیلتر کالمن
LCS	Learning Classifier Systems	سیستم‌های طبقه‌بندی یادگیری
LGP	Linear Genetic Programming	برنامه‌نویسی ژنتیکی خطی
ML	Machine Learning	یادگیری ماشین
MSR	Mean Squared Residence	میانگین مربعات باقیمانده

MPA	Marine Predator Algorithm	الگوریتم شکارچی دریایی
OPSM	Order-Preserving Submatrix	زیرماتریس ترتیب‌حفظ‌کننده
PC	Parking Coordinators	هماهنگ‌کننده‌های پارکینگ
QKP	Quadratic Knapsack Problem	مسئله کوله‌پشتی درجه دوم
QUBIC	QUalitative BIClustering	خوشه‌بندی‌دوگانه کیفی
RBML	Rule-Based Machine Learning	یادگیری ماشین مبتنی بر قواعد
RDD	Resilient Distributed Datasets	مجموعه داده‌های توزیع‌شده مقاوم
RL	Reinforcement Learning	یادگیری تقویتی
RSS	Random Subset Selection	انتخاب زیرمجموعه تصادفی
SA	Simulated Annealing	تبرید شبیه‌سازی‌شده
SAMBA	Statistical-Algorithmic Method for Bicluster Analysis	روش آماری-الگوریتمی برای تحلیل خوشه‌بندی‌دوگانه
SB	Spectral Biclustering	خوشه‌بندی‌دوگانه طیفی
SBS	Static Balanced Sampling	نمونه‌گیری متعادل ایستا
SGP	Stack-based Genetic Programming	برنامه‌نویسی ژنتیکی مبتنی بر پشته
SI	Swarm Intelligence	هوش ازدحامی
SL	Supervised Learning	یادگیری نظارت‌شده
SMA	Slime Muld Algorithm	الگوریتم لجن‌مولد
SMSR	Scaling Mean Squared Residence	مقیاس‌بندی میانگین مجذور باقیمانده
SVC	Support Vector Classifier	طبقه‌بند بردار پشتیبان
SVM	Support Vector Machines	ماشین‌های بردار پشتیبان
TAG	Tree-Adjunct Grammar	گرامر وابسته به درخت
TBS	Topology Based Sampling	نمونه‌گیری مبتنی بر توپولوژی
TS	Tabu Search	جستجوی ممنوعه
TSO	Tree Swarm Optimization	بهینه‌سازی ازدحام درخت
VAR	Bicluster Variance	واریانس خوشه‌بندی‌دوگانه
VE	Virtual Error	خطای مجازی
VNS	Variable Neighborhood Search	جستجوی همسایگی متغیر
WOA	Whale Optimization Algorithm	الگوریتم بهینه‌سازی نهنگ

.

بخش اول

الگوریتم‌های فراابتکاری برای یادگیری ماشین: نظریه ها و دیدگاه‌ها

فصل اول

از الگوریتم‌های فراابتکاری تا برنامه‌نویسی خودکار

اس. الیوچ[1]، ب. جربویی[2] و پ. سیاری[3]

چکیده

الگوریتم‌های فراابتکاری [4] به‌عنوان روش‌هایی شناخته‌شده برای حل کارآمد مسائل بهینه‌سازی پیچیده مطرح می‌باشند. علاوه بر این، تولید خودکار برنامه‌های کامپیوتری با استفاده از الگوریتم‌های فراابتکاری به‌عنوان یک استراتژی جستجو به‌تدریج به یک حوزه پژوهشی فعال تبدیل شده است. این علاقه در ابتدا تحت الگوی برنامه‌نویسی ژنتیک تقویت شد. پس از آن، الگوریتم‌های فراابتکاری، به‌ویژه الگوریتم‌های مبتنی بر جمعیت، توسعه یافتند تا بتوانند برنامه‌های مناسبی را برای مسائل خاص تولید کنند. این مقاله به بررسی تقاطع بین الگوریتمهای فراابتکاری و الگوریتم‌های برنامه‌نویسی خودکار می‌پردازد. در این مقاله، محبوب‌ترین الگوریتم‌های فراابتکاری شرح داده شده‌اند و چارچوبی از اجزا و مراحل مورد نیاز برای توسعه یک مکانیزم برنامه‌نویسی خودکار ارائه شده است. شواهد نشان می‌دهد این نخستین تحقیقی است که به شکلی جامع به برنامه‌نویسی خودکار می‌پردازد. در اینجا هم از مشارکت‌های مؤخر و هم جدید بهره گرفته شده تا نقطه آغازی برای سایر پژوهشگران ایجاد کنیم. هدف تحریک علاقه جامعه پژوهشی و به‌روز نگه داشتن آن‌ها در زمینه

---

[1] S. Elleuch: College of Business and Economics, Qassim University, Buraydah, Kingdom of Saudi Arabia
[2] B. Jarboui: Higher Colleges of Technology, Abu Dhabi, United Arab Emirates
[3] P. Siarry: LISSI, University Paris Est Créteil, Vitry-sur-Seine, France e-mail: siarry@u-pec.fr
[4] Metaheuristics

برنامه‌نویسی خودکار با استفاده از روش‌های فراابتکاری است، به امید اینکه این امر منجر به ظهور مشارکت‌های نوآورانه شود.

۱-۱- مقدمه

پیش از بررسی برنامه‌نویسی خودکار فراابتکاری[۱]، مهم است ابتدا منظور از عبارت "برنامه‌نویسی خودکار[۲]"، اصطلاح "فراابتکاری" و ارتباط میان این دو تشریح گردد.

برنامه‌نویسی خودکار[۳] نوعی تولید کد است که با استفاده از برخی روش‌ها به توسعه‌دهندگان اجازه می‌دهد به‌راحتی کدهایی را با سطح انتزاع بالاتری بنویسند. در ابتدا، این اصطلاح به روش‌هایی اشاره داشت که از اسمبلرها[۴] استفاده می‌کردند و قادر به تولید خودکار کد ماشین[۵] بودند [۳]. درواقع، یک اسمبلر با تبدیل خودکار کد باینری به کد ماشین، وظیفه برنامه‌نویس را تسهیل می‌کرد. با گذر زمان، کامپایلرهای جدید برای زبان‌های نسل‌های بعدی معرفی شدند. امروزه، یک اسمبلر یا کامپایلر دیگر "برنامه‌نویسی خودکار" نامیده نمی‌شود. در حقیقت، برنامه‌نویسی خودکار شامل چندین رویکرد است که یکی از معروف‌ترین آن‌ها، ترکیب برنامه[۶] است [۱۸۹]. ترکیب برنامه، برنامه‌های اجرایی را بر اساس مشخصات موجود و بدون نیاز به دخالت یک کارشناس تولید می‌کند [۱۸۸]. در این زمینه، برنامه‌نویسی خودکار فراابتکاری جای دارد [۱۹۰].

از سوی دیگر، الگوریتمهای فراابتکاری دسته‌ای وسیع از الگوریتم‌ها را شامل می‌شوند که برای حل تقریبی مسائل بهینه‌سازی طراحی شده‌اند [۱۹۲]. آن‌ها از روش‌های خاصی برای جستجو در فضای جستجو و هدایت فرآیند بهینه‌سازی استفاده می‌کنند. به‌طور کلی، هدف الگوریتمهای فراابتکاری به حداقل یا حداکثر رساندن یک تابع هدف است که بر اساس ویژگی‌های مسئله تعریف

---

[1] Metaheuristic Automatic Programming
[2] Automatic Programming
[3] Automatic Programming
[4] assemblers
[5] machine code
[6] program synthesis

می‌شود. در سال‌های اخیر، این الگوریتم‌ها به‌عنوان جایگزین‌های خوبی برای روش‌های کلاسیک، که به روش‌های ریاضی و برنامه‌ریزی پویا متکی می‌باشند، ظاهر شده‌اند. به‌طور کلی، الگوریتمهای فراابتکاری به مسائلی اعمال می‌شوند که با سختی آن‌ها شناخته می‌شوند و هیچ الگوریتم خاص و رضایت‌بخشی برای حل آن‌ها وجود ندارد. این الگوریتم‌ها انعطاف‌پذیر می‌باشند و نیاز به سازگاری عمیق با هر مسئله ندارند. از لحاظ نظری، برخی از آن‌ها به راه‌حل بهینه برخی مسائل با زمان اجرایی مورد انتظار همگرا می‌شوند.

در سی سال گذشته، علاقه به الگوریتمهای فراابتکاری رشد چشمگیری داشته است. با وجود تنوع الگوریتم‌های موجود در این حوزه، مشاهده می‌کنیم که در هر دوره، شاهد ظهور یک فراابتکاری جدید می‌باشیم. برخی از آن‌ها به‌طور کلی برای روش‌های برنامه‌نویسی خودکار تعمیم داده شده‌اند. در یک فراابتکاری کلاسیک، الگوریتم یک راه‌حل یا جمعیتی از رشته‌ها را تکامل می‌بخشد. یک فراابتکاری برنامه‌نویسی خودکار بدون نیاز به تعیین ساختار و شکل راه‌حل از پیش، برنامه‌ها را مدیریت می‌کند. این حوزه با الگوریتم برنامه‌نویسی تکاملی[1] [۱۳] آغاز شد، اما با پیشنهاد جان کوزا[2] درباره الگوریتم برنامه‌نویسی ژنتیک[3] [۱۴] بیشتر شناخته شد.

برنامه‌نویسی ژنتیک با موفقیت در بسیاری از مسائل یادگیری ماشین به کار گرفته شده است [۲۲۴]. درحالی‌که پژوهش‌های علمی درباره برنامه‌نویسی ژنتیک همچنان ادامه دارد، سایر الگوریتمهای فراابتکاری برنامه‌نویسی خودکار نیز ظهور کرده‌اند و در برخی مسائل نتایجی مشابه یا حتی عملکردی بهتر از الگوریتم برنامه‌نویسی ژنتیک ارائه داده‌اند. در سال ۱۹۹۸، تکامل گرامری[4] توسط رایان[5] و همکارانش برای بهینه‌سازی برنامه‌ها معرفی شد [۱۷]. دو سال بعد، روکس[6] و

[1] evolutionary programming
[2] Koza
[3] Genetic Programming
[4] Grammatical Evolution
[5] Ryan
[6] Roux

فونلوپت[۱] اولین روش برنامه‌نویسی خودکار مبتنی بر ازدحام[۲] به نام برنامه‌نویسی مورچه‌ای[۳] را معرفی کردند [۱۶]. برنامه‌نویسی ایمنی[۴] در سال ۲۰۰۳ بر اساس اصول سیستم ایمنی مهره‌داران پیشنهاد شد [۱۸]. در سال ۲۰۰۴، برنامه‌نویسی ازدحام ذرات[۵] و برنامه‌نویسی باکتریایی[۶] به‌ترتیب توسط اونیل[۷] و کابریتا[۸] توسعه یافتند. کلونی زنبور مصنوعی[۹] نیز بعدها به‌عنوان یک روش در برنامه‌نویسی خودکار به کار گرفته شد [۲۱]. در سال ۲۰۱۴، اولین روش برنامه‌نویسی خودکار مبتنی بر جستجوی محلی تکراری[۱۰] توسط نگوین[۱۱] و همکارانش معرفی شد و در زمان‌بندی پویا کارگاه‌های تولید به‌کار گرفته شد [۷۹]. به‌تازگی، الوچ[۱۲] و همکارانش ویژگی‌های اولین کار نوآورانه خود تحت عنوان "برنامه‌نویسی همسایگی متغیر[۱۳] را ارائه کرده‌اند [۸۰].

هوش مصنوعی[۱۴] و یادگیری ماشین[۱۵] اصلی‌ترین حوزه‌های کاربردی برنامه‌نویسی خودکار فراابتکاری می‌باشند. این الگوریتم‌ها قادرند فرآیندهای هوشمندی را متناسب با یک مسئله خاص ایجاد کنند. ازجمله روش‌های سنتی یادگیری ماشین، می‌توان به برنامه‌نویسی منطقی استقرایی[۱۶] [۱۸۰]، شبکه‌های عصبی مصنوعی[۱۷] [۱۸۱]، و یادگیری تقویتی[۱۸] [۱۹۱] اشاره کرد.

تا جایی که می‌دانیم، این نخستین تحقیقی است که به‌طورکلی به برنامه‌نویسی خودکار الگوریتمهای فراابتکاری می‌پردازد. هدف این مقاله ارائه یک دید کلی از الگوریتمهای فراابتکاری

---

[1] Fonlupt
[2] swarm-based
[3] Ant Programming
[4] Immune Programming
[5] Particle Swarm Programming
[6] Bacterial Programming
[7] ONeill
[8] Cabrita
[9] Artificial Bee Colony
[10] Iterated Local Search
[11] Nguyen
[12] Elleuch
[13] Variable Neighborhood Programming
[14] Artificial Intelligence
[15] Machine Learning
[16] Inductive Logic Programming
[17] Artificial Neural Networks
[18] Reinforcement Learning

اصلی و توضیح سازگاری آن‌ها برای توسعه برنامه‌هاست. در اینجا هم از تحقیقات هم قدیمی و هم جدید استفاده شده است تا نقطه شروعی برای دیگر پژوهشگران فراهم گردد. علاوه بر این، یک مقایسه دقیق بین الگوریتمهای فراابتکاری برنامه‌نویسی خودکار و روش‌های سنتی یادگیری ماشین ارائه شده و مزایای استفاده از هرکدام بیان گردیده است.

کار به این صورت سازمان‌دهی شده است: در بخش دوم، به‌طور مختصر الگوریتم‌های فراابتکاری معرفی شده است. در بخش ۱-۳، حوزه برنامه‌نویسی خودکار را با جزئیات، شامل انواع نمایش‌های برنامه و مراحل اصلی پیاده‌سازی یک الگوریتم برنامه‌نویسی خودکار، شرح داده شده است. در همین بخش، روش‌های مختلف ایجاد یک برنامه نیز خلاصه شده است. درنهایت، در بخش ۱-۴، وضعیت فعلی تحقیقات و مسیرهای جالب آینده پژوهش بررسی می‌گردد. چند کلمه نتیجه‌گیری در بخش ۱-۵ ارائه شده است.

۱-۲- الگوریتمهای فراابتکاری

تعداد راه‌حل‌ها در هر تکرار یک معیار اصلی برای طبقه‌بندی الگوریتمهای فراابتکاری است [۲۲۹]. دسته اول شامل الگوریتمهای فراابتکاریی مبتنی بر جمعیت است که در آن الگوریتم تعدادی مشخص از راه‌حل‌ها را تکامل می‌دهد. در دسته دوم، الگوریتم‌ها تنها یک راه‌حل را بهبود می‌دهند و به آن‌ها الگوریتمهای فراابتکاری مبتنی بر راه‌حل گفته می‌شود. در این بخش، هر دو دسته الگوریتمهای فراابتکاری را معرفی کرده و الگوریتم‌های شناخته‌شده در هر یک از این دسته‌ها را شرح داده می‌شود.

۱-۲-۱- الگوریتم‌های فراابتکاری مبتنی بر راه‌حل[1]

الگوریتمهای فراابتکاری مبتنی بر راه‌حل یک راه‌حل فعلی را با اعمال برخی تغییرات بهبود می‌دهند. فرآیند جستجو به‌عنوان مسیری در بین همسایگی‌ها قابل مشاهده است [۲۲]. حرکت از یک راه‌حل به راه‌حل دیگر از طریق یک فرآیند تکراری در فضای جستجو انجام می‌شود. کارایی این دسته از

[1] Solution-Based Metaheuristics

الگوریتمهای فراابتکاری در مسائل مختلف بهینه‌سازی اثبات شده است. در بین الگوریتمهای فراابتکاری مبتنی بر راه‌حل به بررسی تبرید شبیه‌سازی‌شده[۱]، جستجوی ممنوعه[۲]، جستجوی همسایگی متغیر[۳] و «جستجوی محلی تکرارشونده[۴] می‌پردازیم.

۱-۲-۱-۱- تبرید شبیه‌سازی‌شده

روش تبرید شبیه‌سازی‌شده (SA[5]) از مکانیک آماری الگوریتم‌های متروپولیس[۶] الهام گرفته شده است [۲۳]. این روش توسط کرک‌پاتریک و همکارانش معرفی شد [۱۰]. بر اساس روش بازپخت است که توسط فلزکاران برای دستیابی به تعادل حرارتی در هر دما به کار می‌رود. این روش شامل گرم‌کردن و سپس خنک‌کردن آهسته یک ماده تا رسیدن به حالت منجمد و پایدار است (حالت تعادل). در مسائل بهینه‌سازی، انرژی توسط تابع هدف نمایش داده می‌شود و راه‌حل معادل حالت سیستم است.

الگوریتم SA با یک راه‌حل اولیه S که به‌طور تصادفی تولید یا با استفاده از یک روش ابتکاری به دست آمده است، شروع می‌شود. ابتدا دما T مقداردهی اولیه می‌شود. سپس در هر تکرار، یک همسایه S_0 از S به‌طور تصادفی انتخاب می‌شود. حرکت از S به راه‌حل S_0 تنها در صورتی پذیرفته می‌شود که S_0 بهتر از S باشد یا با احتمالی p درصورتی‌که S_0 بدتر از S باشد. تابع هدف f را به‌عنوان تابعی برای کمینه‌سازی مسئله فعلی در نظر بگیرید. احتمال پذیرش بر اساس توزیع بولتزمن[۷] به‌صورت زیر است:

$$p\big(T, f(S_0), f(S)\big) = \exp\left(-\frac{f(S_0) - f(S)}{T}\right) \qquad \text{رابطه (۱-۱)}$$

[1] Simulated Annealing
[2] Tabu Search
[3] Variable Neighborhood Search
[4] Iterated Local Search
[5] Metropolis Algorithms

[7] Boltzmann Distribution

به‌طور مشابه با روش بازپخت، SA به‌تدریج دما T را در طول اجرا کاهش می‌دهد. بنابراین، احتمال پذیرش حرکات تضعیف‌شونده در انتهای جستجو کاهش می‌یابد.

بسیاری از محققان SA را برای حل مسائل بهینه‌سازی گسسته یا پیوسته به کار گرفته‌اند. برای جزئیات بیشتر، می‌توان به کتاب‌شناسی گسترده در [۲۴, ۲۵, ۲۶, ۲۷] مراجعه کرد. در ادبیات علمی، چندین گونه مختلف از SA پیشنهاد شده است [۳۱]، ازجمله روش نویزی [۲۸]، تبرید میکروکانونیک[1] [۲۹]، و روش پذیرش آستانه‌ای[2] [۳۰].

۱-۲-۱-۲- جستجوی ممنوعه

جستجوی ممنوعه (TS[3]) یک الگوریتم فراابتکاری است که در سال ۱۹۸۶ به‌طور رسمی معرفی شد [۱۱]. این روش از تاریخچه یک جستجوی ابتکاری محلی برای فرار از مینیمم‌های محلی و اکتشاف فضای جستجو با استفاده از یک استراتژی از پیش تعیین‌شده استفاده می‌کند. مبنای TS استفاده از حافظه تطبیقی است که از حافظه انسانی الهام گرفته شده است. این حافظه یک رفتار جستجوی انعطاف‌پذیرتر را فراهم می‌کند.

یک لیست ممنوعه، حافظه کوتاه‌مدتی است که در الگوریتم TS برای جلوگیری از تکرار چرخه‌ها به‌کار می‌رود. این لیست ویژگی‌های مسیرهایی را که اخیراً توسط الگوریتم طی شده‌اند، حفظ می‌کند. بنابراین، این حافظه مانع از انتخاب مجدد راه‌حل‌های آخرین بازدید می‌شود. لیست ممنوعه در هر تکرار به‌روزرسانی می‌شود. الگوریتم TS از انواع دیگری از حافظه‌ها نیز استفاده می‌کند: حافظه میان‌مدت و حافظه بلندمدت. حافظه میان‌مدت بهترین راه‌حل‌های یافت‌شده در طول اجرای الگوریتم را ثبت می‌کند. هدف از این مرحله، اکتشاف فضای جستجو حول راه‌حل‌هایی است که دارای ویژگی‌های مشابه با راه‌حل‌های ذخیره‌شده می‌باشند. این مرحله تشدید یا تمرکز[4]

---

[1] Microcanonical Annealing
[2] Threshold Accepting
[3] Tabu Search
[4] Intensification

نامیده می‌شود. حافظه بلندمدت در الگوریتم TS تعریف شده است تا تنوع‌بخشی در فضای جستجو را تضمین کند. درواقع، هدف آن اجبار به جستجوی راه‌حل‌های بازدیدنشده با تشویق حرکت‌هایی بر اساس ویژگی‌های بهترین راه‌حل‌ها است. سپس مجموعه‌ای از قواعد که به‌عنوان معیارهای تمایل[1] شناخته می‌شوند، محدودیت‌های ممنوعه را نادیده گرفته و ممکن است حرکات ممنوع را مجاز کنند. یک بررسی جامع از TS و اصول آن در [۳۲, ۳۳] موجود است.

۱-۲-۱-۳- الگوریتم تپه‌نوردی[2]

الگوریتم تپه‌نوردی یکی از قدیمی‌ترین الگوریتم‌های بهینه‌سازی ریاضی است [۱۸۷]. این روش بر یک جستجوی محلی تکراری استوار است که با یک راه‌حل اولیه شروع می‌شود تا به بهینه‌های محلی دست یابد. درواقع، این الگوریتم همسایگی را بررسی کرده و تنها راه‌حل‌های کاندیدایی را می‌پذیرد که تابع برازش را بهبود دهند.

هنگامی‌که مسئله فعلی کمی پیچیده باشد، صعود تپه‌ای نتایج خوبی ارائه نمی‌دهد. بنابراین، چندین روش در ادبیات علمی برای بهبود فرآیند جستجوی آن پیشنهاد شده است. به‌عنوان مثال، استراتژی پذیرش دیرهنگام [۲۰۳] که شامل مقایسه راه‌حل فعلی با راه‌حلی است که چندین تکرار قبل تولید شده است، یکی از این روش‌ها است.

۱-۲-۱-٤- جستجوی همسایگی متغیر

همان‌طور که در مقدمه اشاره شد، جستجوی همسایگی متغیر (VNS[3]) توسط حسن و املادنکویچ[4] [1] معرفی شد. استراتژی این روش بر اساس تغییر قاعده‌مند در ساختارهای همسایگی استوار است. در ابتدای حل هر مسئله، مجموعه‌ای از ساختارهای همسایگی با تعداد P باید تعریف شوند. سپس

---

[1] Aspiration Criteria
[2] Hill Climbing
[3] Variable Neighborhood Search - VNS
[4] Hansen & Mladenovic

از یک راه‌حل اولیه S، الگوریتم با استفاده از حرکت‌های پیچیده‌تر به‌تدریج به بهینه‌های محلی در تمام ساختارهای همسایگی انتخاب‌شده می‌رسد.

مراحل اصلی الگوریتم VNS عبارت‌اند از :لرزش[۱]، جستجوی محلی[۲] و تغییر همسایگی[۳]. در مرحله لرزش، یک راه‌حل S_0به‌طور تصادفی از همسایگی nام راه‌حل S انتخاب می‌شود. مجموعه ساختارهای همسایگی که در فاز لرزش استفاده می‌شود، می‌تواند با ساختارهای همسایگی جستجوی محلی متفاوت باشد. دو استراتژی جستجوی شناخته‌شده‌ای که به‌عنوان جستجوی محلی به کار می‌روند، به نام‌های بهبود اول[۴] و بهبود بهترین[۵] معروف می‌باشند. در روش بهبود اول، اولین راه‌حل بهبودیافته S' که بهتر از راه‌حل فعلی S است، انتخاب می‌شود. در روش بهبود بهترین، تمامی راه‌حل‌های بهبودیافته انتخاب شده و بهترین آن‌ها (در صورت وجود) به‌عنوان S' انتخاب می‌شود [۳۴].

هدف از مرحله تغییر همسایگی، هدایت همسایگی متغیر است تا مشخص کند در مرحله بعد کدام همسایگی باید جستجو شود و آیا S' به‌عنوان راه‌حل جدید پذیرفته خواهد شد یا خیر. رایج‌ترین روش‌های تغییر همسایگی شامل تغییر همسایگی ترتیبی[۶]، تغییر همسایگی چرخه‌ای[۷]، تغییر همسایگی لوله‌ای[۸] و تغییر همسایگی کج[۹] می‌باشند. خوانندگان علاقه‌مند می‌توانند برای توضیحات بیشتر به منابع [۳۴، ۳۵، ۳۶] مراجعه کنند.

---

[1] Shaking
[2] Local Search
[3] Neighborhood Change
[4] First Improvement
[5] Best Improvement
[6] Sequential Neighborhood Change
[7] Cyclic Neighborhood Change
[8] Pipe Neighborhood Change
[9] Skewed Neighborhood Change

بسیاری از انواع مختلف از طرح‌های اصلی VNS استخراج شده‌اند [۳۴]. ازجمله این موارد می‌توان به جستجوی همسایگی ثابت[۱]، VNS پایه[۲]، VNS عمومی[۳]، VNS کج[۴]، VNS چرخه‌ای[۵]، VNS پیچیده[۶] و VNS دوسطحی[۷] اشاره کرد. این انواع مختلف نشان می‌دهند که ابتکاری‌های VNSمی‌توانند به‌طور موفقیت‌آمیزی برای انواع مختلف مسائل بهینه‌سازی NP-hard به کار گرفته شوند.

۱-۲-۱-۵ جستجوی محلی تکرارشونده

چارچوب جستجوی محلی تکرارشونده (ILS[8]) در سال ۱۹۹۸ توسط استوتزل[۹] [۳۷] تعریف شد. سه سال بعد، این محقق و همکارانش، طرح اصلی ILS را توضیح دادند و آن را برای حل مسئله زمان‌بندی تک‌ماشینی با هدف کمینه کردن دیرکرد وزن‌دار کل به کار گرفتند [۱۳۹]. سپس، لورنزه[۱۰] [۳۸] نسخه‌ای عمومی از این روش ارائه داد. درواقع، به‌جای اینکه در هر تکرار، جستجوی محلی روی یک راه‌حل جدید که به‌طور تصادفی تولید شده اعمال شود (مانند روش جستجوی محلی چندآغازی)، در ILS بهینه محلی به‌دست‌آمده از تکرار قبلی با یک تغییر[۱۱] کوچک اصلاح می‌شود تا یک نقطه شروع جدید برای جستجوی محلی ایجاد گردد. اصل تغییر تأثیر بزرگی بر روند عملکرد الگوریتم ILS دارد. درواقع، اگر این تغییر بسیار ضعیف باشد، ممکن است الگوریتم نتواند از بهینه محلی مشابهی که قبلاً به آن همگرا شده بود، فرار کند. از سوی دیگر، یک تغییر بسیار قوی، الگوریتم را به یک جستجوی محلی با چندین نقطه شروع مختلف تبدیل می‌کند. مرور کاملی از ILS، انواع آن و کاربردهایش در [۳۹] ارائه شده است.

---

[1] Fixed Neighborhood Search
[2] Basic VNS
[3] General VNS
[4] Skewed VNS
[5] Cyclic VNS
[6] Nested VNS
[7] Two-Level VNS
[8] Iterated Local Search
[9] Stutzle
[10] Lourenço
[11] Perturbation

۲-۲-۱- الگوریتم‌های فراابتکاری مبتنی بر جمعیت[1]

تفاوت اصلی میان الگوریتمهای فراابتکاری مبتنی بر جمعیت و الگوریتمهای فراابتکاری مبتنی بر راه‌حل، استفاده از یک مجموعه از راه‌حل‌ها (جمعیت) به‌جای یک راه‌حل واحد است. مفهوم اصلی الگوریتمهای فراابتکاری مبتنی بر جمعیت، بهبود تکراری یک جمعیت از راه‌حل‌ها است. ابتدا، یک جمعیت اولیه از راه‌حل‌ها مقداردهی اولیه می‌شود. در مرحله بعد، راه‌حل‌های جدیدی با استفاده از برخی عملگرها تولید می‌شوند. درنهایت، جمعیت جدید با استفاده از استراتژی‌های انتخاب، از ترکیب جمعیت فعلی و راه‌حل‌های جدید تشکیل می‌شود.

الگوریتمهای فراابتکاری مبتنی بر جمعیت شناخته‌شده را می‌توان به دو دسته تقسیم کرد که شامل محاسبات تکاملی (EC) و هوش ازدحامی (SI) است. اولین الگوریتم‌های توسعه‌یافته در دسته EC بر اساس نظریه تکامل داروین (استفاده از عملگرهای بازترکیب و جهش) پایه‌گذاری شده‌اند. دسته دوم، از الگوهای ساده‌ای از تعامل اجتماعی برای تولید هوش محاسباتی استفاده می‌کند.

۱-۲-۲-۱ محاسبات تکاملی

دسته محاسبات تکاملی (EC^2) به‌عنوان الگوریتم‌های تکاملی (EA^3) نیز شناخته می‌شود. الگوریتم‌های ژنتیک [۴۰] و استراتژی‌های تکامل(ES^4) [۴۱] به EC تعلق دارند و از تکامل طبیعی داروین الهام گرفته‌اند. با این حال، برخی از روش‌های EC مانند الگوریتم تخمین توزیعی (EDA^5) ،[۵] تکامل دیفرانسیلی[۶] (DE^6) و الگوریتم‌های هم‌تکاملی[۴۲] ($CoEA^7$) به‌عنوان الگوریتم‌های تکاملی غیر داروینی شناخته می‌شوند. این الگوریتم‌ها بر اساس یک توزیع تعریف‌شده تکامل می‌یابند. در دو بخش بعدی، جزئیات مربوط به GA ، ES و DE را ارائه می‌دهیم.

---

[1] Population-Based Metaheuristics
[2] Evolutionary Computation
[3] Evolutionary Algorithms
[4] Evolution Strategies
[5] Estimation of Distribution Algorithm
[6] Differential Evolution
[7] Coevolutionary Algorithms

الگوریتم ژنتیک

الگوریتم ژنتیک (GA) شاید مشهورترین و پرکاربردترین روش محاسبات تکاملی باشد. این الگوریتم در اوایل دهه ۱۹۷۰ پدیدار شد [۱۲]. در ابتدا، GAبا یک نمایش باینری از راه‌حل (راه‌حل) مرتبط بود. سپس این نمایش با سایر نمایشها بهبود یافت. GA بر اساس انتخاب[۱]، ترکیب[۲] و جهش[۳] پیاده‌سازی می‌شود که از پدیده‌های تکامل انسانی الهام گرفته شده‌اند.

ترکیب به‌عنوان اصلی‌ترین عامل تغییر شناخته می‌شود. به‌طور کلی، دو کروموزوم منتخب، که به آن‌ها والدین نیز گفته می‌شود، با ترکیب بخشی از هرکدام، راه‌حل‌ها جدیدی را ایجاد می‌کنند. جهش به یک (یا چند) کروموزوم از جمعیت فعلی اعمال می‌شود و آن را به‌طور تصادفی تغییر می‌دهد. این عملگر تضمین‌کننده تنوع ژنتیکی از یک نسل به نسل دیگر است. نقش تابع تناسب[۴] در GA ارزیابی کیفیت هر راه‌حل است. به‌طور معمول، در تکرار بعدی کروموزوم‌هایی که بیشترین تناسب را دارند حفظ می‌شوند. انتخاب راه‌حل یک مرحله مهم در الگوریتم GA است. برخی از روش‌های شناخته‌شده انتخاب شامل انتخاب تورنمنت[۵]، انتخاب بر اساس رتبه‌بندی[۶] و انتخاب چرخه رولت[۷] می‌باشند. یک مطالعه تطبیقی در مورد روش‌های انتخاب استفاده شده در GA توسط گلدبرگ [۴۵] و بلوم [۴۶] انجام شده است.

فرایند تکاملی GA به تعداد پارامترهایی مانند احتمال ترکیب، احتمال جهش، اندازه جمعیت، تعداد نسل‌ها و غیره وابسته است. این پارامترها توسط کاربر تعیین و تنظیم می‌شوند. در مقالات علمی، بیشتر محققان در GA احتمال ترکیب را در محدوده [۰/۱ و ۰/۶] تنظیم کرده‌اند [۴۷] و نرخ

---

[1] Selection
[2] Crossover
[3] Mutation
[4] Fitness Function
[5] Tournament Selection
[6] Ranking Selection
[7] Roulette-Wheel Selection

جهش معمولاً کمتر از ۱ درصد است. درواقع، مقادیر مناسب برای این احتمالات هنوز یک موضوع تحقیقاتی باز است. مرورهای اخیر در [۴۳، ۴۴، ۷۵] برای خوانندگان علاقه‌مند در دسترس است.

استراتژی‌های تکاملی

استراتژی‌های تکاملی (ES[1]) در ابتدا توسط ریچنبرگ و اسچوفل[2] در دانشگاه فنی برلین در سال ۱۹۶۴ معرفی شدند [۴۱, ۴۸]. الگوریتم‌های ES معمولاً برای حل مسائل بهینه‌سازی پیوسته به‌کار می‌روند. اولین کاربرد ES در زمینه بهینه‌سازی پارامترهای تجربی بود. این فرآیند از یک عملگر ساده جهش و انتخاب که به ES[3] دو عضوی معروف است، تشکیل شده است.

در سال‌های اخیر، CMA-ES توسط هانسن[4] و همکارانش به‌عنوان یک گسترش از ES توسعه داده شد [۷]. این روش مؤثر، در حال حاضر به‌طور گسترده‌ای استفاده می‌شود. هانسن تأیید کرده است که CMA-ES در هنگام استفاده برای بهینه‌سازی محلی [۴۹] و بهینه‌سازی جهانی [۵۰] بسیار رقابتی است.

در سال‌های اخیر، کاربرد استراتژی‌های تکاملی بر روی طیف گسترده‌ای از مسائل بررسی شده و در مقالات نظری متعددی [۵۱,۵۲,۵۳,۵۴] خلاصه شده است.

تکامل دیفرانسیلی

همان‌طور که قبلاً ذکر شد، الگوریتم تکامل دیفرانسیلی (DE[5]) توسط استورن[6] و ک.پرایس[7] در سال ۱۹۹۶ معرفی شد [۶]. تکامل دیفرانسیلی درواقع یک فراابتکاری تکاملی مبتنی بر بردار است که به دلیل استفاده از عملگرهای ژنتیکی شباهت‌هایی با الگوریتم‌های ژنتیک دارد.

---

[1] Evolution Strategies
[2] Rechenberg & Schwefel
[3] Two-Membered ES
[4] Hansen
[5] Differential Evolution
[6] Storn
[7] K. Price

درواقع، DE از یک جمعیت راه‌حل‌ها که به آن‌ها عامل[1] نیز گفته می‌شود، شروع می‌کند. استراتژی DE بر اساس سه مرحله ارزیابی، انتخاب و ترکیب استوار است. در فرآیند ترکیب، دو عضو از جمعیت به‌صورت تصادفی انتخاب می‌شوند و سپس راه‌حل جدیدی بر اساس تفاوت بین این دو راه‌حل انتخابی ایجاد می‌شود. بسیاری از توسعه‌های الگوریتم DE برای حل مسائل دنیای واقعی ایجاد شده‌اند. پیشرفت‌های اخیر در زمینه DE در مرجع [۴] در دسترس است.

۱-۲-۲-۲- هوش ازدحامی[2]

این دسته شامل چندین الگوریتم فراابتکاری الهام‌گرفته از هوش ازدحامی است، مانند بهینه‌سازی کلونی مورچه‌ها[3]، بهینه‌سازی ازدحام ذرات[4]، الگوریتم کلونی زنبور مصنوعی[5] و غیره.

بهینه‌سازی کلونی مورچه‌ها

الگوریتم بهینه‌سازی کلونی مورچه‌ها (ACO[6]) در سال ۱۹۹۲ توسط دوریگو[7] پیشنهاد شد [۹]. ایده اصلی آن بر اساس تقلید از رفتار جستجوی غذای کلونی‌های مورچه‌ها بود. درواقع، بینایی مورچه‌های واقعی ضعیف است. بنابراین، این مورچه‌ها برای جستجوی مناطق اطراف و یافتن غذا از راه رفتن تصادفی استفاده می‌کنند. در طول مسیرشان، ماده‌ای شیمیایی به نام فرومون روی زمین می‌گذارند. فرومون (مسیر) به مورچه‌ها کمک می‌کند تا مسیرهای خوب را علامت‌گذاری کنند و دیگران را برای دنبال کردن آن هدایت کنند [۵۵]. یک مورچه به سمت مسیری جذب می‌شود که بیشترین غلظت فرومون را دارد.

مراحل اصلی چارچوب ACO عبارتند از:

-مقداردهی‌اولیه: تمام پارامترهای ACO تنظیم شده و فرومون‌ها مقداردهی‌اولیه می‌شوند.

---

[1] Agent
[2] Swarm Intelligence
[3] Ant Colony Optimization
[4] Particle Swarm Optimization
[5] Artificial Bee Colony Algorithm
[6] Ant Colony Optimization
[7] Dorigo

-ساخت راه‌حل مورچه: برای ساخت راه‌حل‌های اولیه، مورچه‌های مصنوعی از یک راه‌حل جزئی خالی شروع می‌کنند که با اضافه کردن یک جزء ممکن از زیرمجموعه‌ای از اجزای ممکن که می‌توان اضافه کرد، درحالی‌که امکان‌پذیری حفظ می‌شود، گسترش می‌یابد. این گسترش بر اساس یک انتخاب احتمالی انجام می‌شود.

-به‌روزرسانی فرومون: شامل دو مرحله است. مرحله اول تبخیر فرومون است که شامل کاهش غلظت فرومون در اجزا در طول زمان می‌باشد. این فرآیند به‌منظور جلوگیری از همگرایی به راه‌حل‌های بهینه‌ی محلی انجام می‌شود. مکانیزم دوم، رسوب فرومون است که برای جذاب‌تر کردن اجزای راه‌حل‌های با کیفیت بالا اعمال می‌شود [۵۶].

-عمل نظارتی: این اقدام باید توسط بیش از یک مورچه انجام شود. اعمال یک جستجوی محلی معمولاً رایج‌ترین عمل نظارتی است.

-الگوریتم ACO به‌طور موفقیت‌آمیز به چندین مسئله بهینه‌سازی ازجمله مسیریابی، زمان‌بندی و تخصیص منابع اعمال شده است [۵۶].

بهینه‌سازی ازدحام ذرات

بهینه‌سازی ازدحام ذرات (PSO[1]) یک الگوریتم هوش ازدحامی است که توسط کندی و همکاران [۸] معرفی شد. این الگوریتم از رفتار اجتماعی گروهی پرندگان که در جستجوی غذا می‌باشند، الهام گرفته است.

درPSO ، یک جمعیت شامل N راه‌حل کاندید به نام ذرات[2] تشکیل می‌شود. ذرات به‌طور محلی و بدون کنترل مرکزی با یکدیگر هماهنگ می‌شوند. هر ذره با یک بردار سرعت، موقعیتی در فضای جستجو و حافظه‌ای که بهترین موقعیت قبلی‌اش را ذخیره می‌کند، مشخص می‌شود. ذرات با بهترین کیفیت، بر رفتار کلونی تأثیر می‌گذارند. درواقع، به‌روزرسانی بردار سرعت بر اساس سرعت قبلی

[1] Particle Swarm Optimization
[2] particles