

بررسی الگوریتم های بهینه سازی مبتنی بر هوش گروهی

اعظم گلستان^۱، زینب مولای زاهدی^۲

^۱ دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی واحد بروجرد

^۲ دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات فارس، باشگاه پژوهشگران جوان و نخبگان

a_golestan90@yahoo.com

F_zahedi2008@yahoo.com

کد مقاله: Com-1035

چکیده

بهینه سازی، یکی از حوزه های تحقیقاتی مهم در دهه های اخیر بوده است که نتیجه آن طراحی انواع مختلفی از الگوریتم ها بوده است. در بسیاری از مسائل مهندسی معمولاً با تابع هدفی روبه رو هستیم که می خواهیم آن را بهینه کنیم. در روش های گروهی، عامل ها با هم همکاری می نمایند و رفتار جمعی تمام عامل ها باعث یک همگرایی می شود. حال روش های گوناگونی برای رسیدن به این نقطه بهینه وجود دارد که مسلماً هر کدام از این روش ها معایبی دارند. در این مقاله الگوریتم های هوش گروهی مانند الگوریتم کلونی مورچه ها، الگوریتم کرم شب تاب و غیره مورد مقایسه قرار گرفته است.

واژه کلیدی: بهینه سازی - هوش گروهی - الگوریتم کلونی مورچه ها

۱. مقدمه

مفهوم بهینه سازی^۱ بدین صورت است که در بین پارامترهای یک تابع به دنبال مقادیری باشیم که تابع را کمینه یا بیشینه نماید. کلیه مقادیر مناسب جهت این امر، راه حل های ممکن و بهترین مقدار از این مقادیر را، راه حل بهینه می نامند. الگوریتم های بهینه سازی هر دو نوع مسائل بیشینه سازی و کمینه سازی را پوشش می دهند. بهینه سازی کاربرد های زیادی در زمینه تخصیص منابع، زمان بندی ها، تصمیم گیری ها و ... را دارد. بهینه سازی همواره با مشکلات فراوانی همراه بوده است. شیوه های سابق برای حل کردن مشکلات بهینه سازی، مستلزم تلاش های محاسباتی بیشماری می باشد. الگوریتم هایی از جمله الگوریتم های هوش جمعی^۲ تا حدی این مشکل را حل نموده اند. توسط این الگوریتم ها راه حل هایی پیدا می شوند که تقریباً به جواب نزدیکند [۱]. هوش جمعی نوعی روش هوش مصنوعی^۳، مبتنی بر رفتار های جمعی می باشد. عامل ها، به طور محلی با یکدیگر و با محیط خود در تعامل هستند. موفق ترین روش های هوش جمعی که تاکنون به وجود آمده اند، روش بهینه سازی کلونی مورچه^۴ ها، روش بهینه سازی اجتماع ذرات^۵، روش بهینه سازی زنبور عسل^۶ و روش بهینه سازی کرم شب تاب^۷ می باشد.

1. Optimization

2. Swarm Intelligence

3. Artificial Intelligence

4. Ant Colony Optimization

5. Particle Swarm Optimization

بیشتر الگوریتم های تکاملی الهام خود را از طبیعت دریافت می کنند تا روابط پیچیده، را از شرایط اولیه و ساده درک نمایند. طبیعت مثال کامل و مطلوبی برای بهینه سازی است. هر الگوریتم با ایجاد یک جمعیت اولیه از راه حل های کاربردی آغاز می گردد و به صورت مکرر از نسلی به نسلی دیگر در جهت بهترین راه حل حرکت می کند. در تکرار های موفق الگوریتم، انتخاب در مجموعه ای از راه حل ها انجام می گیرد [۲]. این مقاله در ۱۰ بخش، بازمینی و تحقیق در مورد الگوریتم های بهینه سازی، در راستای حوزه های کاربردی مربوطه را ارائه می نماید. در بخش های ۲ و ۳ و ۴ این مقاله الگوریتم های تکاملی و بخش های ۵ تا ۹ برخی از الگوریتم های هوش جمعی ارائه شده است. بخش ۱۰ این مقاله نیز نتیجه گیری می باشد.

۲. الگوریتم های ژنتیک^۸

ژنتیک یک الگوریتم بهینه سازی احتمالی، با یک پتانسیل جستجوی سراسری می باشد که در سال ۱۹۷۵ توسط Holland ارائه شده است. الگوریتم های ژنتیک جزو موفق ترین دسته از الگوریتم های تکاملی می باشند. از الگوریتم ژنتیک به عنوان یک بهینه ساز تابع یاد شده است. این الگوریتم با مقداردهی اولیه یک مجموعه راه حل (کروموزوم^۹) آغاز می شود. شامل ارائه مجدد مشکل و معمولاً به شکل یک بردار کوچک می باشد. سپس به ازای هر کروموزوم، تناسب را با استفاده از یک تابع تناسب مناسب برای مشکل ارزیابی می کنند. بر این اساس، بهترین کروموزوم ها در استخراج جفتگیری انتخاب می شوند، در آنجا آنها دچار برخورد شده و جهش صورت می گیرد، سپس مجموعه جدیدی از راه حل ها ارائه می شوند. (زاد و ولد).

الگوریتم های ژنتیک در فضاهایی که جستجو تا حد زیادی پیچیده و ناشناخته می باشند مفید و موثر می باشد. حال اگر تابع تناسب به خوبی تعریف نشده باشد، ممکن است الگوریتم ژنتیک، ماکزیمم محلی را به جای ماکزیمم سراسری انتخاب نماید. در این الگوریتم عملیات بر اساس مجموعه داده های پویا دشوار می باشد و در بعضی موارد الگوریتم های ساده تر بهتر از ژنتیک به جواب دست پیدا می کنند. [۱]

۳. الگوریتم استراتژی تکاملی^{۱۰}

یک نمونه دیگر در خانواده الگوریتم های تکاملی، الگوریتم استراتژی تکاملی (DE) می باشد که توسط Price و Storn در سال ۱۹۹۵ ارائه شد. این الگوریتم مشابه ژنتیک می باشد زیرا از جمعیت های افراد برای جستجو به دنبال یک راه حل بهینه استفاده می شود. تفاوت اصلی میان ژنتیک و استراتژی تکاملی، این است که در ژنتیک جهش نتیجه اختلالات کوچکی در ژن های یک فرد می باشد، درحالیکه در تکاملی، جهش نتیجه جفت گیری های حسابی افراد می باشد. در ابتدای فرآیند تکامل، عملگر جهش الگوریتم تکاملی، تمایل به پوشش دارد. زمانیکه تکامل پیشرفت می کند، عملگر جهش به بهره کشی تمایل دارد. بنابراین، این الگوریتم به صورت خود کار ترقی های جهش به بهترین مقدار مبتنی بر مرحله فرآیند تکاملی را می پذیرد. جهش در الگوریتم های تکاملی، مبتنی بر یک تابع چگالی احتمال از پیش تعریف شده نمی باشد. این الگوریتم تکاملی ساده اما دقیق می باشد. در این الگوریتم کاربر باید، بهترین مقادیر را برای پارامترهای کنترل وابسته به مشکل بیابد و این یک وظیفه زمانبر می باشد و جزء عیوب این الگوریتم به حساب می آید. [۴]

6. Artificial Bee Colony Algorithm

7. Firefly Algorithm

8. Genetic Algorithm

9. Chromosome

10. Differential Evolution

۴. الگوریتم شالیزار برنج^{۱۱}

الگوریتم شالیزار برنج (PFA) در سال ۲۰۰۹ توسط Premaratne و سایرین ارائه شد که بر طبق یک اصل نزدیکی به راه حل اصلی را عمل می کند. برخلاف الگوریتم تکاملی، دربرگیرنده رفتار مرکب و تقاطع میان افراد نمی باشد و در عوض از گرده افشانی و پراکنده سازی استفاده می کند. این الگوریتم شامل پنج گام اصلی می باشد.

- تخمکاری: این الگوریتم با پراکنده سازی بذرها به شکل تصادفی (جمعیت اولیه) در یک زمین ناهموار عمل می کند.
- انتخاب: در این مرحله بهترین گیاهان براساس یک شیوه آستانه، برای حذف انتخابی راه حل های نامطلوب انتخاب می شوند و جمعیت نیز کنترل می گردد.
- تولید بذرها: در این قسمت هر گیاه تعدادی از بذرها را متناسب با سلامت خود تولید می کند. این بذرها بر روی اغلب مکان های مطلوب می افتند (حاصلخیز ترین خاک، بهترین زهکشی، رطوبت خاک مطلوب و غیره) و تمایل دارند تا بهترین گیاهان باشند (بلندتر) و تعداد بذرهای بیشتری را تولید کنند. بالاترین گیاه این جمعیت به مکانی با شرایط بهینه متعلق بوده و تناسب گیاه براساس یک تابع تناسب مشخص می شود.
- گرده افشانی: برای پخش بذر، گرده افشانی به واسطه حیوانات یا به واسطه باد یک عامل عمده می باشد. تراکم بالای جمعیت، شانس گرده افشانی را برای گرده حمل شده توسط باد افزایش خواهد داد.
- پراکنندگی: برای ممانعت از تلمبار شدن با حداقل مقدار محلی، بذرهای هر گیاه پراکنده می شوند. بسته به موقعیت زمین، گیاهان جدیدی رشد خواهند کرد و این چرخه ادامه می یابد. الگوریتم شالیزار برنج دارای هزینه محاسباتی کمتری می باشد. از آنجاییکه دارای تقاطع نمی باشد، راه حل بهینه می تواند مهاجرت کردن به سمت بهترین ها باشد.

۵. الگوریتم بهینه سازی اجتماع ذرات

بهینه سازی ازدحام ذرات (PSO) یک تکنیک بهینه سازی سراسری مبتنی بر جمعیت، ارائه شده توسط Kennedy و Eberhart در سال ۱۹۹۵ می باشد. این روش از رفتار اجتماعی پرندگان برای جستجوی خوراک الهام گرفته شده است. به دلیل مکانیسم ساده جستجو، کارآیی محاسباتی و پیاده سازی آسان، به صورت گسترده ای در بسیاری از حوزه های بهینه سازی اعمال می شود. در الگوریتم اجتماع ذرات، واژه ذرات به اعضای جمعیتی اشاره دارد که حجم کم و جرم کمی دارند (یا دارای یک جرم یا حجم قراردادی کوچک می باشند). هر ذره در ازدحام نشان دهنده یک راه حل در فضایی با بعد بالا همراه با چهار بردار می باشد. موقعیت جاری آن، بهترین موقعیت یافت شده در دور، بهترین موقعیت یافته شده توسط همسایگی دورش و سرعتش این چهار بردار را شامل می گردد. موقعیت هر ذره در فضای جستجو براساس بهترین موقعیت به دست آورده شده توسط خودش (pbest) و بهترین موقعیت دستیابی شده توسط همسایگی اش (gbest) در طی فرآیند جستجو تعیین می گردد. در هر تکرار، هر ذره موقعیت و سرعت خود را به صورت زیر به روز رسانی می کند

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (1)$$

$$v_{k+1}^i = v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i)$$

¹¹ .Paddy Field Algorithm

در اینجا، x نشان دهنده موقعیت ذره و v نشان دهنده سرعت آن می باشد. r_1, r_2 نشان دهنده اعداد تصادفی میان صفر و یک می باشد و C_1, C_2 نشان دهنده پارامترهای شناختی می باشد. [۱۰]

۵-۱. برتری الگوریتم اجتماع ذرات نسبت به الگوریتم ژنتیک :

- پیاده سازی الگوریتم اجتماع ذرات ساده تر می باشد و پارامترهای کمتری برای تنظیم وجود دارند.
- الگوریتم اجتماع ذرات دارای یک قابلیت حافظه موثرتری نسبت به الگوریتم ژنتیک می باشد.
- الگوریتم اجتماع ذرات در حفظ تنوع ازدحام کارآمدتر می باشد، زیرا تمامی ذرات از اطلاعات مربوط به موفق ترین ذره برای بهبود خود استفاده می کنند، اما در الگوریتم ژنتیک، راه حل های بدتر رد می شوند و تنها راه حل های جدید ذخیره می گردند (زیرمجموعه ای از بهترین ها را دربر می گیرد).
- شبهت های زیادی میان الگوریتم اجتماع ذرات و الگوریتم ژنتیکها وجود دارند. هردوی آنها راه حل ها را مقدار دهی اولیه می کنند و نسل ها را به روز رسانی می کنند، درحالیکه الگوریتم اجتماع ذرات هیچ عملگر تکاملی ندارد. این الگوریتم علاوه بر توابع پیوسته، نشان دهنده ثبات و تقارب در یک فضای پیچیده نیز می باشد.

۶. الگوریتم بهینه سازی کلونی مورچه

الگوریتم کلونی مورچه جزء موفق ترین الگوریتم های مبتنی بر ازدحام می باشد، که در سال ۱۹۹۹ توسط Di Caro و Dorigo ارائه شده است. این الگوریتم یک متا ذهنی الهام شده توسط رفتار کاوشی مورچه ها در محیط می باشد. جالب ترین اشتراک در رفتار مورچه ها، توانایی آنها برای یافتن مسیرهای فرعی میان لانه و منابع غذا توسط پیمایش ردهای فرومون می باشد. سپس مورچه ها با تصمیم احتمالی مبتنی بر میزان فرومون^{۱۲}، مسیری را برای ادامه انتخاب می کنند. هرچه رد فرومون قویتر باشد، مطلوبیت آن مسیر نیز بیشتر خواهد بود. چون مورچه ها به دلیل فرومون باقیمانده بروی مسیر، آن مسیر را پیگیری می کنند، این رفتار منجر به هدایت فرآیند خودیاری برای تشکیل مسیرهای مشخص شده توسط غلظت بالای فرومون می شود. با مدلسازی و شبیه سازی رفتار کاوشی مورچه، مرتب سازی تخم ها، لانه سازی و ذخیره سازی و غیره، الگوریتم هایی را می توان توسعه داد، که برای مشکلات بهینه سازی پیچیده و ترکیبی قابل استفاده باشند. اولین الگوریتم مورچه با نام سیستم مورچه^{۱۳} (AS) توسط Dorigo و سایرین در سال ۱۹۹۶ توسعه داده شده و به صورت موفق برای ارزیابی مقایسه ای مشکل مسافرت فروشنده، آزموده شد. [۹]

الگوریتم کلونی مورچه در سه تابع اصلی به صورت زیر طبقه بندی می شود :

- AntSolutions Construct : این تابع فرآیند ساخت راه حل ها را در جایی انجام می دهد، که مورچه ها به واسطه وضعیت های مجاور یک مشکل، باید برطبق یک قانون انتقال حرکت کنند.
- Pheromone Update: این تابع دربرگیرنده به روز رسانی ردهای فرومون می باشد. به روز رسانی پس از هر تکرار انجام می گردد. علاوه بر تقویت رد فرومون، الگوریتم کلونی مورچه، دربرگیرنده تبخیر رد فرومون نیز می باشد. تبخیر ردهای فرومون به مورچه ها کمک می کند تا راه حل های بدی را که قبلاً آموخته اند، را فراموش کنند.

¹² . Pheromone

¹³ . Ant System

- Deamon Actions: این یک گام اختیاری در الگوریتم است، که دربرگیرنده اعمال به روز رسانی های افزوده از یک دیدگاه سراسری می باشد. این تابع ممکن است شامل اعمال تقویت افزوده فرومون برای بهترین راه حل ایجاد شده باشد.

۷. الگوریتم کلونی مصنوعی زنبور (ABC)

الگوریتم های هوش ازدحامی مختلفی براساس رفتار زنبورها در طبیعت، موجود می باشند. این الگوریتم ها به دو دسته طبقه بندی می شوند؛ رفتار کاوشی و رفتار جفتگیری. الگوریتم مصنوعی کلونی زنبور (ABC)، رفتار کاوشی زنبورها را شبیه سازی می کند. یک نهاد فردی (مانند یک زنبور در یک کلونی زنبور) نشان دهنده مجموعه ساده ای از سیاست های رفتاری (مانند مهاجرت، دفاع و مرگ) می باشد ولی یک گروه از نهادها (مانند یک کلونی زنبور) نشان دهنده رفتار پیشامدی پیچیده ای با ویژگی های مفید مانند مقیاس پذیری و سازگار پذیری است.

کلونی مصنوعی زنبور یک الگوریتم شبیه سازی کننده رفتارهای کاوشی و هوشمندانه، مجموعه ای از زنبورهای عسل می باشد که توسط Karaboga و Basturk ارائه گردید. در این الگوریتم، زنبورها شامل زنبورهای کارگر^{۱۴}، تماشاچی^{۱۵} و کاوشگر^{۱۶} می باشد. [۷]

یک زنبور منتظر جهت تصمیم گیری برای انتخاب یک منبع غذا، زنبور تماشاچی نامیده می شود و زنبوری که منبع غذا توسط او نظارت می شود، زنبور کارگر است. نوع دیگری از زنبورها، زنبور کاوشگر می باشد که جستجوی تصادفی را برای کشف منابع جدید انجام می دهد. موقعیت یک منبع نشان دهنده یک راه حل ممکن برای مشکل بهینه سازی بوده و میزان شربت منبع (کیفیت غذا) به ارزش راه حل اضافه می گردد. یک ازدحام از زنبورهای مجازی تولید شده و جستجو به صورت تصادفی آغاز می گردد. زنبورها تا زمانی تعامل دارند، که شربت نهایی را بیابند و راه حل مشکل از طریق شدت تعامل های این زنبورها به دست آورده می شود.

یک زنبور کارگر، راه حل ها را در حافظه خود و بسته به اطلاعات محلی (اطلاعات بصری) تولید می کند و میزان شربت (مقدار تناسب) منبع را بررسی می کند. با توجه به این موضوع که میزان شربت منبع جدید بالاتر از منبع قبلی باشد، زنبور موقعیت جدید را به خاطر سپرده و قبلی را فراموش می کند. پس از اینکه تمامی زنبورهای کارگر فرآیند جستجو را کامل نمودند، آنها اطلاعات شربت منابع (غذا) و اطلاعات موقعیت آنها را، برای مورچه های کاوشگر به اشتراک می گذارند.

در فاز بعدی بر اساس میزان احتمال مربوط به منبع غذا یعنی P_i ، زنبور کاوشگر یک منبع غذا را انتخاب می کند.

$$P_i = \frac{food_i}{\sum_{n=1}^N food_n} \quad (2)$$

در اینجا، N تعداد منابع خوراکی است، $food_i$ مقدار تناسب محلول i است که با میزان شربت منبع خوراکی در موقعیت آن متناسب است.

در آخرین فاز (تعویض زنبور و انتخاب)، اگر یک موقعیت را نتوان بهبود داد، آنگاه آن منبع خوراکی به صورت رد شده در نظر گرفته می شود. مقدار تعداد از پیش مشخص شده چرخه ها یک پارامتر کنترلی مهم از الگوریتم کلونی زنبور می باشد که حدی برای رها سازی نامیده می شود. پس از هر عمل،

^{۱۴} . Employed Bees

^{۱۵} . Onlookers

^{۱۶} . Scouts

موقعیت منبع تولید می شود و سپس توسط زنبور مصنوعی ارزیابی می شود و کارآیی آن با کارآیی نمونه قدیمی مقایسه می شود. اگر خوراکی جدید دارای شربت بهتر یا مساوی با منبع قدیمی باشد، جایگزین نمونه قدیمی در حافظه می شود. در غیر اینصورت، منبع قدیمی در حافظه بازاریابی می شود.

کارآیی جستجوی محلی الگوریتم کلونی زنبور به جستجوی همسایگی و مکانیسم های انتخاب حریصانه توسط زنبورهای کارگر و کاوشگر بستگی دارد. کارآیی جستجوی سراسری الگوریتم نیز به فرآیند جستجوی تصادفی انجام شده توسط زنبورهای کاوشگر و مکانیسم تولید راه حل مجاور توسط زنبورهای کاوشگر و کارگر وابسته می باشد.

۸. الگوریتم کرم شب تاب

الگوریتم کرم شب تاب عنوان الگوریتم ذهنی مبتنی بر ازدحام، برای وظایف بهینه سازی محدود، توسط Yang ارائه شد. در این الگوریتم از رفتار تابشی کرم های شب تاب الهام گرفته شده است. این الگوریتم یک رویه تکراری مبتنی بر جمعیت را با عوامل بشمار (تحت عنوان کرم های شب تاب) به کار می گیرد. به این عوامل امکان داده می شود تا فضای تابع هزینه را به صورت موثرتری نسبت به جستجوی تصادفی توزیع شده، بررسی کنند. تکنیک بهینه سازی هوشمند، مبتنی بر این فرضیه است که راه حل یک مشکل بهینه سازی را، می توان به عنوان عاملی (کرم شب تاب) در نظر گرفت که به صورت متناسب با کیفیت آن در یک محیط تابیده می شود. متعاقباً هر کرم شب تاب، همتایان خود را (صرف نظر از جنسیتشان) جذب می کند که فضای جستجو را به صورت موثرتری بررسی می کند.

الگوریتم کرم شب تاب دارای سه قانون خاص می باشد که مبتنی بر برخی ویژگی های کرم های شب تاب واقعی است. این سه قانون عبارتند از:

- تمامی کرم های شب تاب دو جنسیتی هستند و آنها صرف نظر از جنسیت خود به صورت جذاب تر و شفاف تری حرکت خواهند کرد.
- درجه جذابیت یک کرم شب تاب با درخشش آن متناسب است. همچنین ممکن است درخشندگی با افزایش فاصله از کرم های شب تاب دیگر کاهش یابد. حال اگر یک کرم شب تاب جذاب تر یا شفاف تری نسبت به این کرم وجود نداشته باشد، آنگاه به صورت تصادفی حرکت خواهد کرد.
- درخشندگی یا شدت نور یک کرم شب تاب، توسط مقدار تابع هدف مشخص می شود. [۳]

این الگوریتم مبتنی بر برقراری ارتباط سراسری میان ذرات می باشد. بنابراین در بهینه سازی چند هدفی کارآمد تر و موثرتر است.

۹. الگوریتم سیستم دفاعی مصنوعی^{۱۷}

این الگوریتم در سال ۱۹۹۹ توسط Dasgupta ارائه شد. الگوریتم دفاع مصنوعی مبتنی بر اصل انتخاب غیر جنسیتی بوده و یک الگوریتم مبتنی بر جمعیت می باشد. الگوریتم سیستم دفاعی مصنوعی از سیستم ایمنی انسان الهام گرفته شده است. یک سیستم سازگار توزیع شده دارای توانایی های تشخیص مصنوعی، آموزش تقویت، استخراج خصیصه، حافظه ایمنی، تنوع و قدرتمندی می باشد. سیستم دفاعی مصنوعی توانایی ها را ترکیب نموده است.

قدرت جستجوی اصلی در سیستم دفاعی مصنوعی متکی بر عملگر جهش می باشد. بنابراین عامل تصمیم گیری در این تکنیک نیز می باشد. گام های این الگوریتم به صورت زیر تعریف می گردد:

- مقاداردهی اولیه پادتن ها (راه حل های بالقوه برای مشکل) در مرحله نخست انجام می گیرد. آنتی ژن ها نشان دهنده مقدار تابع هدف $f(x)$ برای بهینه شدن می باشند.

¹⁷. Artificial Immune System Algorithm

- تولید مثل، که در آن تناسب یا نزدیکی هر پادتن مشخص می شود. براساس این تناسب، پادتن ها تکثیر می شوند؛ که در بهترین حالت بیشترین تکثیر صورت می گیرد. تعداد کلون های ایجاد شده از طریق n پادتن به صورت زیر به دست می آید: [6]

$$NC = \sum \text{round} \left(\beta * \frac{i}{i} \right) \quad i = 1, 2, 3, \dots, n \quad (3)$$

- در اینجا NC تعداد کل کلون ها می باشد، β یک ضریب تکثیر کننده و j اندازه جمعیت پادتن ها می باشد.
- ابر جهش، گام آخر این الگوریتم می باشد. کلون ها به یک فرآیند ابر جهش مرتبط می شوند که در آن کلون ها در تناسب معکوس برای وابستگی، جهش داده می شوند؛ بهترین کلون های پادتن، کمتر جهش می یابند و کلون های دارای پادتن بدتر، بیشترین جهش را دارند. سپس کلون ها در راستای پادتن های اصلی خود ارزیابی می شوند که بهترین پادتن ها برای تکرار بعدی انتخاب می شوند. جهش می تواند یکنواخت، گاوس یا نمایی باشد.

۱۰. نتیجه گیری

الگوریتم های تکاملی نوعی جستجوی تصادفی بر مبنای جمعیت می باشند که اولین و مهمترین نقطه قوت آن ها این است که ذاتاً موازی اند، و می توانند فضای جستجو را در جهات مختلف بررسی نمایند. بررسی موازی زیر فضاها باعث می شود که جستجوی فضا به نواحی که متوسط آماری تابع هدف در آن ها زیاد است و امکان وجود نقطه بهینه مطلق در آن ها بیشتر می باشد، سوق پیدا کند. چون در این روش ها بر خلاف روش های تک مسیری، فضای جواب به طور همه جانبه جستجو می شود، امکان کمتری برای گرفتار شدن در یک نقطه بهینه محلی وجود خواهد داشت. الگوریتم های تکاملی روش هایی مستقل از مسیر هستند. این موضوع آن ها را نسبت به روش های قدیمی متمایز می سازد، لذا می توانند به صورت موثری تمام فضای جستجو را بررسی کنند. همچنین الگوریتم های تکاملی وابستگی به پیوستگی تابع و یا مشتق آن را ندارند، و در روند جستجو تنها به تعیین مقدار تابع هدف در نقاط مختلف نیاز است. این الگوریتم ها فقط به اطلاعاتی در مورد کیفیت راه حل های ایجاد شده به وسیله هر مجموعه از متغیرها نیاز دارند، در صورتی که روش های دقیق نیاز به شناخت کاملی از ساختمان مسئله و متغیرها دارند.

الگوریتم های دیگری نیز در سال های اخیر نوشته شده اند و سعی در بهبود این الگوریتم ها داشته اند. اما در اینجا برخی از الگوریتم های بهینه سازی، بطور مختصر توضیح داده شد.

منابع

- [1] Shilane. D, Martikainen. J. S and Dudoit. S, A general frame-work for statistical performance comparison of evolutionary computation algorithms, Information Sciences, Journal, 178, 2870-2879 , 2008.
- [2] Vallada. E and Ruiz. R, Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem, Omega , 38 , 57 -67, 2011.
- [3] Yang. X.S, Firefly algorithm for multimodal optimization in proceedings of the stochastic Algorithms, Foundations and Applications(SAGA 109) , 5792 of Lecture notes in Computer Sciences Springer,2009.
- [4] Chen. H and Zhu. Y, Optimization based on symbiotic multi-species coevolution, journal on Applied Mathematics and Computation , 2008.
- [5] Storn. R and Price K, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11, 341–359, 2012.
- [6] Mehrabian. A.R and Lucas. C, A novel numerical optimization algorithm inspired from weed colonization, Ecological Informatics 1 355–366, 2006.
- [7] Karaboga. D and Basturk. B, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39, 459–471, 2009.
- [8] Sayadi. M.K and Ramezani. R, A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems, International Journal of Industrial Engineering Computations, 1-10, 2010.

- [9] Darigo. M and Stutzle. T, Ant Colony Optimization IEEE Computational Intelligent Magazine, Vol. 1, 28-39, 2008.
- [10] Kennedy. J and Eberhart. R.C, Particle Swarm Optimization, IEEE International Conference on Neural network, 1, 192-194, 2005.