



دانشگاه کاشان
University of Kashan

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir



ارائه یک الگوریتم ترکیبی با استفاده از الگوریتم کرم شب تاب، الگوریتم ژنتیک و جست و جوی محلی*

جواد سلیمی سرتختی^{۱*}، استادیار، سلمان گلی بیدگلی^۲، استادیار
^۱ دانشکده مهندسی برق و کامپیوتر، گروه مهندسی کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

چکیده

تاریخچه مقاله:

دریافت ۱۹ اردیبهشت ماه ۱۳۹۹
پذیرش ۰۳ شهریور ماه ۱۳۹۹

کلمات کلیدی:

الگوریتم تکاملی کرم شب تاب چندهدفه
الگوریتم ژنتیک
جست و جوی محلی
بهینه سازی پیوسته

در این مقاله، یک الگوریتم ترکیبی چندهدفه ارائه شده است که ویژگی های دو الگوریتم ژنتیک و کرم شب تاب را ترکیب می کند. این الگوریتم با مجموعه ای از کرم های شب تاب که در فضای مسئله به صورت تصادفی پخش می شوند، شروع به کار می کند و این ذرات طی مراحل تکامل، به جواب بهینه مسئله همگرا می شوند. سپس یک طرح جست و جوی محلی به عنوان روشی برای جست و جوی همسایگی به منظور بهبود کیفیت جواب ها ارائه و پیاده سازی شده است. این بخش از الگوریتم برای جست و جوی نواحی کم جمعیت، برای یافتن جواب های غالب استفاده می شود. برای بهبود الگوریتم تغییراتی در معیار تعیین بهترین بهینه سراسری به ازای هر کرم شب تاب و همچنین بهترین بهینه محلی اعمال شده است. استفاده از این روش موجب شده یکنواختی منحنی پرتو بیشتر شود. نتایج آزمایشگاهی روش ارائه شده بر روی برخی از توابع محک نشان می دهد که به کارگیری این روش موجب کاهش خطا شده است. الگوریتم پیشنهادی بر اساس یک الگوریتم پایه توسعه داده شده است.

© ۱۳۹۹ - مجله محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

بهینه سازی چندهدفه، همواره موضوع تحقیقاتی مهم برای محسوب شده است. در این گونه مسائل، مجموعه ای از راه حل های غلبه ناپذیر به جای یک راه حل تولید می شوند. به اصطلاح یک راه حل غلبه ناپذیر است، در صورتی که هدفی نتواند بدون بدتر شدن حداقل یک هدف دیگر بهبود یابد [۱]. به طور کلی دو نوع برخورد با این گونه مسائل وجود دارد: ۱. روش های چندهدفی سنتی که سعی می کردند تا مجموعه ای از راه حل های غلبه ناپذیر را با استفاده از برنامه ریزی ریاضیاتی

پیدا کنند. درباره مسائل غیرخطی، روش وزن دهی و روش محدودیت E از تکنیک های متداول هستند [۲]. هر دو روش مسئله را به مسئله تک هدفه تبدیل می کنند، به گونه ای که بتوان با استفاده از بهینه سازی غیرخطی، مسئله را حل نمود. با روش وزن دهی، اگر تمامی وزن ها مثبت باشند، راه حل های غلبه ناپذیر به دست می آیند، اما تمامی راه حل های بهینه پرتو را نمی توان یافت مگر اینکه تمامی توابع هدف مانند ناحیه قابل قبول، محدب باشند. نقطه ضعف دیگر این روش این است که بسیاری از مجموعه های مختلف وزن ها، ممکن است همان راه حل را تولید کنند و موجب افت بهره وری شوند. زمانی که وزن ها ترجیحات تصمیم گیرنده را نشان می دهند، روش مورد نظر بهترین سازش را در اختیار ما قرار می دهد. برای مثال، راه حلی که بالاترین سوددهی از تصمیم گیرنده را تولید کند. در

* نوع مقاله: پژوهشی

* نویسنده مسئول

پست های الکترونیک: salimi@kashanu.ac.ir (سلیمی سرتختی)

goli@kashanu.ac.ir (گلی بیدگلی)

نحوه ارجاع به مقاله: سلیمی سرتختی، جواد، گلی بیدگلی، سلمان، «ارائه یک الگوریتم ترکیبی با استفاده از الگوریتم کرم شب تاب، الگوریتم ژنتیک و جست و جوی محلی»، مجله محاسبات نرم، جلد ۸، شماره ۱، ص ۲۸-۱۴، بهار و تابستان ۱۳۹۸.

تک هدفه مبتنی بر گرادیان و الگوریتم بهینه سازی چندهدفه تکاملی است. این روش از نقاط مرجع به عنوان محدودیت ها استفاده می کند تا الگوریتم را قادر سازد که مسئله بهینه سازی غیرمحدب را حل کند. بهینه ساز مبتنی بر گرادیان به عنوان عملگر جهش الگوریتم تکاملی و طرح مورد استفاده به منظور سوئیچ بین جهش ژنتیکی و جهش مبتنی بر گرادیان پیشنهاد شده است. برنامه ریزی تکاملی چندهدفه (MOEP)، با استفاده از حاصل جمع رتبه فضایی با انتخاب متنوع در مرجع [۸] معرفی شده است. این روش، از روش MOEP کلاسیک در مراحل مرتب سازی و انتخاب متفاوت است. حاصل جمع رتبه به منظور تقسیم هر هدف به صد رتبه و جمع متناظر رتبه بندی برای تمام اهداف استفاده شده است. به جمعیت، حاصل جمع رتبه مطابق با موقعیت آن در فضای جست و جو، تخصیص داده شده است. در مرجع [۹]، یک پروسه تکمیل شده جست و جوی محلی مبتنی بر EMO ارائه شده است که در آن، روش NSGA2 به عنوان الگوریتم EMO استفاده شده است و این روش با تابع اسکالرزسازی ترکیب شده است. جست و جوی محلی از راه حل فرزندان که به عنوان نقاط مرجع در گرفته می شوند، آغاز می شود. جست و جوی محلی از نقاط مرجع استفاده کرده و حداقل به یک راه حل بهینه پرتو محلی نزدیک به نقاط مرجع دست می یابد.

الگوریتم بهینه سازی ذرات از مشاهده رفتار اجتماعی حیوانات مانند پرندگان و ماهی ها الهام گرفته شده است. در مقایسه با الگوریتم ژنتیک، روش بهینه سازی ذرات، ویژگی های جالب توجهی دارد [۱۰]. این روش از حافظه استفاده می کند، به طوری که اطلاعات راه حل های خوب توسط تمامی ذرات، حفظ می شود، در حالی که در الگوریتم ژنتیک، دانش قبلی از مسئله با تغییر جمعیت، دور اندخته می شود. همچنین این روش، همکاری سازنده بین ذرات دارد؛ زیرا ذرات در جمعیت، اطلاعات را میان خودشان مبادله می کنند. بهبود عملکرد PSO، شامل سه مرحله است: ۱. گسترش فضای جست و جوی صحیح، تنظیم پارامترها و ترکیب با دیگر روش ها [۱۱]. از آنجایی که PSO نمی تواند به طور مستقیم

روش محدودیت E نیازی به تحذب نیست، اما اگر شرایط خاصی ارضا شوند، تنها به یک راه حل غلبه ناپذیر منجر می شود [۲]. ۲. الگوریتم های تکاملی با توجه به توانایی های بالا در جست و جوی هم زمان برای چند راه حل پرتو و انجام جست و جوی جهانی بهتر در فضای جست و جو، نامزدهای خوبی برای حل مسائل چندهدفه می باشند [۳].

از طرفی الگوریتم ژنتیک از بهترین گزینه ها برای مسائل چندهدفه هستند. الگوریتم ژنتیک، یک روش جست و جوی تصادفی بر اساس مکانیزم انتخاب، تولید مثل و تکامل می باشد [۴]. از آنجایی که این الگوریتم به طور هم زمان، بسیاری از جواب ها را در فضای جست و جو و ارزیابی می کند، به احتمال زیاد می تواند راه حل جهانی مسئله را می تواند پیدا کند (هرچند تضمینی برای آن وجود ندارد). پیاده سازی و استفاده از روش های الگوریتم ژنتیک آسان است. الگوریتم های فراابتکاری کاربرد فراوانی در مباحث مربوط به بهینه سازی چندهدفه دارند. آن ها قاعدتاً از ویژگی های موفقی که در طبیعت وجود دارد الهام گرفته اند [۵]. برای مثال الگوریتم های چندهدفه ژنتیک بسیار شناخته شده هستند. الگوریتم های چندهدفه تکاملی تفاضلی ابزارهای قدرتمندی برای این نوع مسائل هستند. در کنار این دو، الگوریتم های چندهدفه بهینه سازی ذرات نیز بشدت مورد استقبال قرار گرفته اند. در ادامه به برخی از این الگوریتم ها اشاره می شود.

اولین پیاده سازی الگوریتم ژنتیک چندهدفه، الگوریتم ژنتیک ارزیابی برداری نام دارد. پس از آن، الگوریتم های تکاملی زیادی برای حل مسائل چندهدفه گسترش یافته اند. به تازگی، تأکید بسیاری بر بهبود الگوریتم های تکاملی به منظور دستیابی به محاسبات کارآمد و همگرایی مناسب، صورت گرفته است. در مرجع [۶]، یک الگوریتم تکاملی خود تطبیقی تفاضلی با استراتژی یادگیری هوشمند هدف ارائه شده است. پارامترهای باز ترکیبی و جهش مناسب، در مسائل بهینه سازی چندهدفه، برای هر هدف به طور جداگانه آموزش داده می شود. یک الگوریتم ژنتیک میکرو بر اساس بایگانی ترکیبی در مرجع [۷] پیشنهاد شده است. این روش ترکیبی از بهینه سازی

مختلف با استفاده از وزن‌دهی به آن‌ها را دارند؛ از این دست الگوریتم‌ها می‌توان به کار ارائه‌شده توسط آپوستلیولوس و ولاچس اشاره کرد [۱۶]. ۲. الگوریتم‌هایی که الگوریتم کرم شب‌تاب را به‌منظور تولید جبهه پرتو بهینه گسترش داده‌اند که در این زمینه هم می‌توان به کارهای [۱۷-۱۹] اشاره کرد.

برخلاف کارهای ارائه‌شده تاکنون، در این مقاله، یک روش ترکیبی جدید با استفاده از الگوریتم کرم شب‌تاب، الگوریتم ژنتیک و جست‌وجوی محلی ارائه شده است. در روش پیشنهادی علاوه بر تغییراتی که در الگوریتم کرم شب‌تاب ایجاد شده، یک روش جدید به‌منظور انجام جست‌وجوی محلی در اطراف جواب‌های بالقوه ارائه شده است. در این روش، از یک جواب بالقوه شروع می‌کنیم، اطراف آن را بر اساس یک هیورستیک خلاقانه جست‌وجو می‌کنیم و بعد به سراغ جواب بالقوه دیگر می‌رویم. این کار را تا زمانی انجام می‌دهیم که یا محدودیت زمانی لازم برای جست‌وجوی محلی نقض شود یا به جواب مدنظر برسیم.

ادامه این مقاله بدین صورت است: در بخش ۲ مسئله بهینه‌سازی چندهدفه به‌طور مفصل تعریف شده است؛ بخش ۳ شامل مروری بر الگوریتم‌های بهینه‌سازی کرم شب‌تاب و ژنتیک است و در بخش ۴ رویکرد پیشنهادی به‌طور کامل مورد بحث قرار گرفته است. بخش ۵ شامل نتایج آزمایشگاهی است و در بخش آخر نیز مقاله جمع‌بندی و کارهای آینده معرفی شده است.

۲. بهینه‌سازی چندهدفه

یک مسئله بهینه‌سازی که شامل M هدف است، به‌صورت رسمی با عبارت زیر می‌تواند بیان شود (در نظر داشته باشید که مسائل بهینه‌سازی در این مقاله از نوع کمینه‌سازی هستند):

$$\text{Minimize } \vec{f}(\vec{x}) = [f_i(\vec{x}), i = 1, 2, \dots, M] \quad (1)$$

$$g_j(\vec{x}) \leq 0, j = 1, 2, \dots, J$$

که در آن، $\vec{x} = [x_1, x_2, \dots, x_n]$ n بُعد فضای متغیر تصمیم‌گیری، M نشان‌دهنده ابعاد فضای هدف، $f_i(\vec{x})$ نشان‌دهنده i امین تابع هدف، و $g_j(\vec{x})$ J امین محدودیت

به‌منظور بهینه‌سازی چندهدفه به کار رود، دو موضوع به‌هنگام استفاده از PSO در مسائل چندهدفه می‌بایست در نظر گرفته شود. موضوع اول، چگونگی انتخاب بهترین ذره محلی و جهانی برای هدایت جست‌وجوی ذرات می‌باشد و موضوع دوم چگونگی حفظ نقاط مناسبی می‌باشد که تاکنون یافت شده‌اند.

به‌تازگی، یانگ یک الگوریتم فراابتکاری جدید با نام کرم شب‌تاب^۱ ارائه کرده است [۱۲]. الگوریتم کرم شب‌تاب از برخی ویژگی‌های موجود در حشره‌های کرم شب‌تاب الهام گرفته شده است. یک کرم شب‌تاب تمایل دارد به سمت کرم‌های شب‌تاب با قدرت تابش بیشتر جذب شود. این الگوریتم دو تفاوت و برتری نسبت به الگوریتم بهینه‌سازی ذرات دارد: ۱. جاذبه‌های محلی، ۲. گروه‌بندی مجدد خودکار. هرچه شدت نور با فاصله کم شود، بسته به ضریب جاذبه، جذابیت در بین کرم‌های شب‌تاب می‌تواند محلی یا جهانی باشد و به این ترتیب همه حالت‌های محلی و همچنین حالت‌های جهانی بررسی می‌شوند. همچنین کرم‌های شب‌تاب می‌توانند با توجه به قدرت جذب همسایگانشان به زیرگروه‌هایی تقسیم و دوباره گروه‌بندی شوند. بنابراین هر زیرگروه در این راهکار، توده‌ای از ذرات حول یک نقطه بهینه محلی خواهد بود. ویژگی دوم این الگوریتم را به‌شدت برای بهینه‌سازی‌های چندهدفه مناسب می‌سازد [۱۳ و ۱۴]. مطالعات انجام‌شده نشان می‌دهد که این الگوریتم، قدرت فراوانی در مباحث بهینه‌سازی دارد و می‌تواند از الگوریتم‌هایی همانند بهینه‌سازی ذرات پیشی بگیرد. برای مثال الگوریتم کرم شب‌تاب (Linde-Buzo-Gray) LGB در برخی کاربردها سرعت بیشتری از الگوریتم‌های مشابه همانند الگوریتم بهینه‌سازی ذرات دارد [۱۵].

در زمینه الگوریتم‌های چندهدفه مبتنی بر الگوریتم کرم شب‌تاب، تاکنون کارهای زیادی انجام نشده است. به‌طور کلی کارهای انجام‌شده را می‌توان به دو دسته تقسیم کرد: ۱. الگوریتم‌های کرم شب‌تابی که سعی بر ترکیب اهداف

نامساوی است. مسئله بهینه‌سازی چندهدفه، منجر به یافتن بردار \vec{x} به‌گونه‌ای می‌شود که تابع $f_i(\vec{x})$ را بهینه نماید. از آنجایی که مفهوم جواب بهینه، در بهینه‌سازی چندهدفه با بهینه‌سازی تک‌هدفه، متفاوت است، مفهوم غلبه در پرتو می‌بایست برای ارزیابی راه‌حل‌ها مورد استفاده قرار گیرد.

تعریف ۱: «غلبه پرتو». برای یک مسئله مینیمم‌سازی (رابطه ۱) چندهدفه، بردار $\vec{U} = [U_1, U_2, \dots, U_M]$ بر بردار $\vec{V} = [V_1, V_2, \dots, V_M]$ غلبه می‌کند (غلبه کردن \vec{U} بر \vec{V} با عبارت $\vec{U} \preceq \vec{V}$ نشان داده می‌شود)، اگر و تنها اگر [۱]:

$$\forall i \in \{1, \dots, M\}, \quad u_i \leq v_i \wedge \exists i \in \{1, \dots, M\}: u_i \ll v_i \quad (2)$$

تعریف ۲: «بهینگی پرتو». راه‌حل $\vec{u} \in U$ (درحالی‌که U شامل همه مقادیر ممکن باشد)، بهینه پرتو است اگر و تنها اگر راه‌حل دیگری مانند $\vec{v} \in U$ وجود نداشته باشد که بردار \vec{u} توسط بردار \vec{v} مغلوب شود. در این حالت، راه‌حل \vec{u} راه‌حل غلبه‌ناپذیر نامیده می‌شود. مجموعه تمامی راه‌حل‌های غلبه‌ناپذیر، مجموعه بهینه پرتو را به دست می‌آورد.

۳. مروری بر الگوریتم‌های بهینه‌سازی کرم شب‌تاب و ژنتیک

۳.۱. بررسی الگوریتم بهینه‌سازی گروهی ذرات

الگوریتم کرم شب‌تاب را یانگ در سال ۲۰۰۸ با الگوبرداری از تابش نور در حشره‌های شب‌تاب ارائه کرد [۱۲]. این الگوریتم یک الگوریتم هوش جمعی است. هر فرد (که در این الگوریتم کرم شب‌تاب نامیده می‌شود) در جمعیت بیانگر یک راه‌حل بالقوه در یک فضای چندبعدی است. به دلیل قدرت جذب بین کرم‌های شب‌تاب، آن‌ها می‌توانند به سمت بقیه مکان‌ها حرکت کنند و راه‌حل‌های بهتری را بیابند. در الگوریتم کرم شب‌تاب، میزان جاذب بودن بر اساس قدرت تابش نور و روشنایی آن فرد تعیین می‌شود؛ این میزان در اصل متناسب با میزان شایستگی یک فرد است. یک مسئله پیدا کردن بیشینه را در

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad \text{where } r_{ij} = \|X_i - X_j\| \quad (3)$$

که در آن، $\|X_i - X_j\|$ فاصله اقلیدسی دو کرم شب‌تاب i و j و پارامتر β_0 میزان جاذب بودن در فاصله صفر را بیان می‌کند. γ نیز ضریب جذب نور را بیان می‌کند. طبق مقاله [۱۳] بهترین مقدار برای γ برابر $\frac{1}{12}$ است که در آن Γ مقیاس طول برای متغیرهای طراحی شده خواهد بود.

برای هر کرم شب‌تاب X_i ، در مقایسه با کرم X_j ، اگر X_j روشنایی بیشتری نسبت به X_i داشته باشد، X_i بر اساس میزان جاذب بودن X_j به سمت X_j حرکت خواهد کرد. میزان این حرکت بر اساس معادله (۴) به دست می‌آید [۱۲].

$$x_{id}(t+1) = x_{id}(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha \varepsilon_i \quad (4)$$

که در آن، x_{id} مؤلفه d م فرد i ام در جمعیت، d بعد راه‌حل، α ضریب اهمیت به حرکت تصادفی، ε_i یک عدد تصادفی و t شماره نسل می‌باشند. الگوریتم شکل (۱) نحوه کار الگوریتم شب‌تاب را بیان می‌کند. خط سوم الگوریتم یک حلقه را نشان می‌دهد که تا زمان برآورده شدن شرط خاتمه، این حلقه تکرار می‌شود. در خطوط بعدی به‌ازای هر دو کرم شب‌تاب در الگوریتم میزان جاذب بودن و میزان حرکت آن‌ها به سمت یکدیگر، طبق رابطه (۴) محاسبه می‌شود. سپس میزان شایستگی هر یک از کرم‌ها محاسبه می‌شود. این کار آنقدر تکرار می‌شود تا شرط خاتمه در خط سه الگوریتم ارضا شود.

```

/* Population initialization */
1 Randomly initialize the population and generate  $ps$  fireflies (solutions)  $X_i(t)$ , where  $i = 1, 2, \dots, ps$  and  $t = 0$ ;
2 compute the fitness value of each firefly;
3 while  $t \leq G_{max}$  do
4   for  $i = 1$  to  $ps$  do
5     for  $j = 1$  to  $ps$  do
        /* Movement through the attraction */
6       if  $f(X_j(t)) < f(X_i(t))$  then
7         Move  $X_i(t)$  towards  $X_j(t)$  according to Eq. 3;
8         Calculate the fitness value of  $X_i(t)$  (new candidate solution);
9       end
10    end
11  end
12  t++;
13 end

```

شکل (۱): شبه کد الگوریتم کرم شب تاب [۱۲]

به کار می‌کنند. هر دو الگوریتم، از مقادیر شایستگی (جاذب بودن) برای ارزیابی جمعیت استفاده می‌کنند و هر دو، جمعیت را به‌روز کرده و برای یافتن جواب بهینه از تکنیک‌های تصادفی استفاده می‌کنند. البته در هر دو روش، رسیدن به جواب نهایی تضمین نمی‌شود. از طرف دیگر، الگوریتم کرم شب تاب، از بازترکیبی و جهش که در الگوریتم ژنتیک استفاده می‌شود، استفاده نمی‌کند و راه‌حل بهینه را با جمعیتی که بهترین کرم شب تاب را دنبال می‌کند (هم به صورت محلی و هم به صورت سراسری) می‌یابد. همچنین، الگوریتم کرم شب تاب، پیش‌زمینه هوش اجتماعی بیشتری دارد و به‌آسانی می‌تواند پیاده‌سازی شود. کرم‌ها در الگوریتم کرم شب تاب، یکدیگر را به‌روز می‌کنند و همچنین این روش از حافظه استفاده می‌کند. بنابراین دانش موجود از راه‌حل‌های مناسب، توسط تمامی کرم‌ها حفظ می‌شود در حالی که در الگوریتم ژنتیک، دانش قبلی از مسئله با تغییر جمعیت دور انداخته می‌شود.

۱. یک جمعیت اولیه ایجاد کن.
۲. میزان شایستگی تمامی اعضای جمعیت را محاسبه کن.
۳. کارهای زیر را تا رسیدن به شرط خاتمه تکرار کن.
 ۱. ۳.۱. از بین اعضای جمعیت والدین را انتخاب کن.
 ۲. ۳.۲. فرزندان را با استفاده از عملگرهای ترکیب و جهش تولید کن.
 ۳. ۳.۳. شایستگی فرزندان را محاسبه کن.
 ۴. ۳.۴. جمعیت بازماندگان را انتخاب و جایگزین کن.

- خاتمه

شکل (۲): شبه کد الگوریتم ژنتیک [۲۰]

۲.۳. الگوریتم ژنتیک

الگوریتم ژنتیک در سال ۱۹۶۰ توسط Holland ارائه و به‌وسیله Glodberg توسعه داده شد. الگوریتم ژنتیک در مسائل بهینه‌سازی مختلف همانند شبکه‌های عصبی با موفقیت مواجه شده است [۲۰]. در هر مسئله، جواب را به صورت کروموزوم در نظر می‌گیرد. سپس یک جمعیت اولیه از افراد (کروموزوم‌ها) تشکیل می‌شود که این جمعیت بخشی از فضای پاسخ مسئله را نشان می‌دهد. قبل از شروع جست‌وجو، مجموعه‌ای از کروموزوم‌ها به صورت تصادفی از فضای پاسخ مسئله به‌عنوان جمعیت اولیه تولید می‌شود. افراد جمعیت بر اساس میزان شایستگی‌شان که از طریق یک تابع هدف تعیین می‌شود، با هم رقابت می‌کنند. عملگرهای ژنتیک مانند انتخاب، جهش و بازترکیبی بر روی افراد جمعیت اعمال شده و در نهایت جمعیت جدیدی از کروموزوم‌ها که برازندگی آن‌ها بهتر از جمعیت قبل است تولید می‌شود. این روند تا زمانی که شرط پایان برقرار شود، ادامه می‌یابد و در نهایت بهترین فرد در جمعیت به‌عنوان جواب انتخاب می‌شود. در شکل (۲) شبه کد مربوط به این الگوریتم نمایش داده شده است.

۳.۳. مقایسه بین الگوریتم ژنتیک و بهینه‌سازی گروهی

ذرات

الگوریتم کرم شب تاب نقاط مشترکی با الگوریتم ژنتیک دارد. هر دو الگوریتم با گروهی از جمعیت تصادفی تولیدشده شروع

برای بهینه سازی چندهدفه معرفی می شود و الگوریتم آن در این حالت MOFA نام می گیرد. در این بخش، فرایند الگوریتم MOFA توصیف می شود که شامل مراحل زیر است:

مرحله ۱: مقداردهی اولیه پارامترهای الگوریتم کرم شب تاب، مقداردهی تصادفی N کرم با وضعیت $X_i^{t=0}$ که در آن $i=1, \dots, N$ خواهد بود.

مرحله ۲: شناسایی مجموعه محلی ($L_i^{t=0}$) برای هر کرم به صورت $L_i^{t=0} = \{X_i^{t=0} | i = 1, \dots, N\}$. همچنین، شناسایی عنصر محلی مطلوب ($LP_i^{t=0}$) از آمین ذره به طوری که:

$$\forall \text{ Firefly } i \exists LP_i^{t=0} = \{X_i^{t=0}\}$$

مرحله ۳: جمع آوری تمام مجموعه های محلی $= \forall i$

$C = \cup_{i=1}^N L_i^{t=0}$ که به گونه ای C در مجموعه $\{L_i^{t=0}\} 1, \dots, N$ باشد.

مرحله ۴: تعریف کردن مجموعه سراسری $G^{t=0} = ND(C)$ ، که $(.)$ به تابعی با قابلیت تشخیص تمام راه حل های غیر غالب اشاره می کند.

مرحله ۵: در فضای هدف، فاصله بین $X_i^{t=0} = 1, \dots, N$ و اعضای $G^{t=0}$ توسط فاصله اقلیدسی و یا سایر معیارهای فاصله اندازه گیری می شود. نزدیک ترین عضو G^t به آمین ذره به صورت عنصر مطلوب سراسری برای GP_i^t تعریف می شود.

مرحله ۶: تعیین مجموعه $E^{t=0}$ معادل با $G^{t=0}$.

مرحله ۷: به روزرسانی وضعیت کرم های شب تاب؛ به روزرسانی موقعیت X_i^t تحت میزان ضریب جذب و تابندگی بر اساس معادلات زیر:

$$x_{id}(t+1) = x_{id}(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha(t) \epsilon_i \quad (5)$$

$$\text{Where } \alpha(t+1) = (1 - \frac{t}{G_{max}}) \alpha(t)$$

مرحله ۸: تکامل کرم های شب تاب؛ برای محدود کردن حرکت یک کرم شب تاب بر اساس میزان تابندگی و جاذب بودن، عامل انقباضی اصلاح شده ارائه شده است که قابل قبول بودن وجود کرم ها حفظ شود. شکل (۳) حرکت کرم نام در فضای جست و جو بدون هیچ عامل کنترلی (خط چین) با یک عامل انقباضی اصلاح شده (خط توپر) را نشان می دهد. جایی

در مقایسه با الگوریتم ژنتیک، مکانیزم مبادله اطلاعات در الگوریتم کرم شب تاب کاملاً متفاوت است. در الگوریتم ژنتیک، کروموزوم ها اطلاعات را با یکدیگر مبادله می کنند، بنابراین تمامی جمعیت مانند یک گروه به سمت ناحیه بهینه حرکت می کند. حال آنکه در الگوریتم کرم شب تاب، کرم های با بهترین تابندگی به سایر کرم ها اطلاعات می دهد و تکامل تنها به دنبال یافتن بهتر راه حل می گردد. در مقایسه با الگوریتم ژنتیک، در الگوریتم کرم شب تاب، کرم ها تمایل به همگرایی به بهترین و سریع ترین راه حل دارند که در بیشتر حالات این راه حل، یک جواب محلی می باشد.

۴. رویکرد پیشنهادی

روش پیشنهادی دقیقاً بر مبنای یک الگوریتم ارائه شده ترکیبی تشکیل شده از الگوریتم بهینه سازی ازدحام ذرات و الگوریتم ژنتیک است [۲۱] که با تغییراتی در آن سعی شده از الگوریتم کرم شب تاب به جای بهینه سازی ازدحام ذرات استفاده شود. تحقیقات نشان داده است که الگوریتم کرم شب تاب در مقایسه با بهینه سازی ذرات و ژنتیک برای پیدا کردن نقاط بهینه عمومی در برخی از کاربردها، از کارایی بیشتری برخوردار است [۲۲ و ۲۳]. بنابراین روش پیشنهادی یک الگوریتم مرکب، حاصل از ترکیب الگوریتم های GA و کرم شب تاب به منظور بهبود کارایی هر دو الگوریتم ارائه می دهد. همچنین برای بهبود کیفیت جواب ها، یک طرح جست و جوی محلی^۱ به منظور جست و جوی همسایگی استفاده می شود، این بخش از الگوریتم برای جست و جوی نواحی کم جمعیت، برای یافتن جواب های غالب^۲ استفاده می شود. الگوریتم ارائه شده از سه مرحله (FA, GA, LS) تشکیل شده است. در ادامه، هرکجا از واژه الگوریتم پایه استفاده شده، منظور روش ارائه شده در مرجع [۲۱] است.

۴.۱. الگوریتم چندهدفه کرم شب تاب

در این بخش، نحوه عملکرد الگوریتم چندهدفه کرم شب تاب

1. Local Search Schema (LS)
2. Non-dominated

شکست خورده افزایش می یابد. موقعیت اصلاح شده جدید کرم به صورت زیر محاسبه می شود:

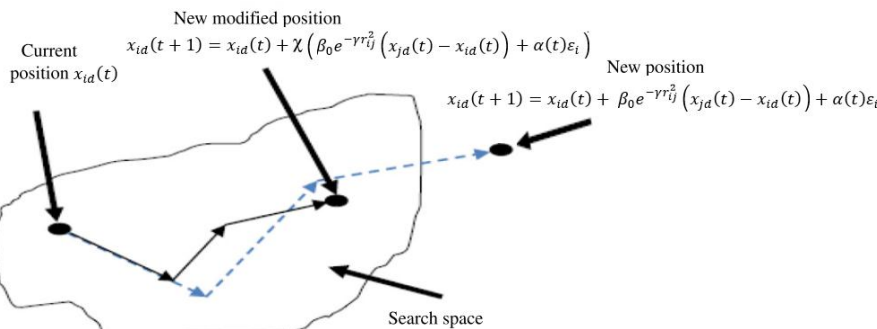
$$x_{id}(t+1) = x_{id}(t) + \chi(\beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha(t)\varepsilon_i) \quad (7)$$

قابل قبول بودن هر کرم شبتاب چک می شود، اگر غیرقابل قبول بود، پارامتر χ برای کنترل موقعیت آن مطابق الگوریتم شکل (۴) اجرا می شود.

که کرم i با موقعیت X_i^t تحت تابندگی و جاذب بودن خاصی در فضای قابل قبول شروع به حرکت کرده، موقعیت جدید کرم شبتاب را زیر سؤال می برد، بنابراین یک عامل اصلاح شده انقباضی χ به صورت زیر تعریف می شود [۲۱]:

$$\chi = \frac{2}{|-2 - \tau - \sqrt{\tau^2 + \tau}|} \quad (6)$$

به صورتی که τ عمر کرم شبتاب غیرقابل قبول (برای مثال طول زمانی که کرم امکان وجود ندارد) و مقدار آن برای حفظ قابل قبول بودن کرم با افزایش آزمایش های



شکل (۳): حرکت ذره i درون فضای جست و جو با استفاده از عامل اصلاح شده انقباضی

```

Input  $(x(t), x(t+1), \alpha, \beta)$ 
while  $x(t+1)$  is infeasible
    Generate new  $x_{id}(t+1) = x_{id}(t) + \chi(\beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha(t)\varepsilon_i)$ 
end
Update  $\alpha(t+1)$ 
Output  $(x(t+1), \alpha(t+1))$ 
    
```

شکل (۴): شبه کد الگوریتم مربوط به اصلاح وضعیت کرم شبتاب با توجه به عامل اصلاح شده انقباضی

مطلوب سراسری GP_i^{t+1} برای هر ذره؛ در فضای هدف، فاصله بین $X_i^{t+1} \forall i = 1, \dots, N$ و اعضای L_i^{t+1} توسط معادله مرحله ۵ اندازه گیری می شود. نزدیک ترین ذره L_i^{t+1} به آمین ذره به صورت LP_i^{t+1} تعریف می شود. همچنین، فاصله بین $X_i^{t+1} \forall i = 1, \dots, N$ و اعضای G_i^{t+1} نیز توسط معادله مرحله ۵ اندازه گیری می شود. نزدیک ترین عضو در G_i^{t+1} به آمین ذره به صورت GP_i^{t+1} مطلوب سراسری تعریف می شود.

۲.۴. الگوریتم ژنتیک

در این بخش، الگوریتم ژنتیک به کاررفته در این مقاله توضیح داده می شود. مراحل این الگوریتم بدین شکل است:

مرحله ۹: به روزرسانی مجموعه L_i^t ؛ موقعیت جدید هر ذره X_i^{t+1} به L_i^t اضافه می شود تا L_i^{t+1} تشکیل شود که به روزرسانی آن بر اساس الگوریتم ۲ خواهد بود.

مرحله ۱۰: به روزرسانی مجموعه سراسری $G: G^{t+1} = ND(U_{i=1}^N L_i^{t+1})$ ، که تمام راه حل های غیرغالب L_i^{t+1} را شامل می شود.

مرحله ۱۱: به روزرسانی مجموعه خارجی E^t : کپی کردن اعضای G^{t+1} در E^t . معیار غلبه برای خارج کردن تمامی پاسخ های غالب E^t اعمال می شود (برای مثال هر عضو G^{t+1} دارای ۳ احتمال خواهد بود).

مرحله ۱۲: به روزرسانی ذره مطلوب محلی LP_i^{t+1} و ذره

مرحله ۶: استراتژی حذف؛ دو فرزند جدید تولید شده جایگزین دو عضو بدتر جمعیت می شوند.

مرحله ۷: ترمیم؛ جواب های غیر معتبر جمعیت باید به جواب های معتبر تبدیل شوند.

۳.۴. مرحله LS

برای بهبود جواب ها یک طرح جست و جوی محلی اعمال می شود، به طوری که به جست و جوی نواحی کم جمعیت در آرشو پرتوهای بهینه تمایل دارد تا اینکه بتواند جواب های غالب بیشتری بیابد. در این بخش، روند اعمال LS را بررسی می کنیم، که یک روش تغییر یافته Hooke و Jeeves می باشد و بر اساس مقاله [۲۱] نوشته شده است. در مرحله نخست با یک نقطه $E^t \in R^n$ ($x_m \in R^n$) و طول گام مشخص Δx_i در جهت ابعاد مختلف، $i = 1, \dots, n$ شروع می کنیم. سپس k ($k = 1, \dots, k_{max}$) تعداد دفعاتی است که این روند باید تکرار شود تا جوابی بهتر x_m حاصل شود. در مرحله سوم، وضعیت x_i حول نقطه فعلی x_m تغییر می کند تا به نقطه x'_m با شرایط زیر دست یابیم:

$$x'_m = \begin{cases} x_m + \Delta x_i & \text{if } f^+(\cdot) > f \\ x_m - \Delta x_i & \text{if } f^-(\cdot) > (f(\cdot) \wedge f^+(\cdot)) \\ x_m & \text{if } f(\cdot) > (f^+(\cdot) \wedge f^-(\cdot)) \end{cases} \quad (11)$$

$\forall i = 1, 2, \dots, n$

فرض می کنیم $f(\cdot)$ مقدار تابع هدف در یک نقطه خاص است، به طوری که $f^+(\cdot) = f(x_m + \Delta x_i)$ ، $f(\cdot) = f(x_m)$ و $f^-(\cdot) = f(x_m - \Delta x_i)$ با مختصر توضیحی کاملاً واضح می شود. سه حالت زیر را در نظر بگیرید:

۱. همان گونه که در شکل (۵) نشان داده شده است، میزان شایستگی در نقطه $X + \Delta x_i$ بیش از نقطه X است. بنابراین اگر از نقطه X به سمت نقطه $X + \Delta x_i$ حرکت کنیم، می توانیم میزان شایستگی این فرد از جمعیت را افزایش دهیم.

۲. شکل (۶) نشان می دهد که میزان شایستگی در نقطه $X - \Delta x_i$ بیش از نقطه X است. بنابراین اگر از نقطه X به سمت نقطه $X - \Delta x_i$ حرکت کنیم، می توانیم میزان شایستگی این فرد از جمعیت را افزایش دهیم.

۳. در نهایت همان گونه که در شکل (۷) ملاحظه می کنید،

مرحله ۱: مقداردهی اولیه جمعیت: در این مرحله، اعضای جمعیت PSO به عنوان جمعیت اولیه برای الگوریتم ژنتیک در نظر گرفته می شود.

مرحله ۲: ارزیابی جمعیت: یک روش برای نگاشت مقادیر توابع هدف به fitness function برای هر رشته ورودی در فضای ژنوتایپ، ترکیب چند تابع هدف به یک مقدار اسکالر به شکل رابطه (۸) می باشد:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_m f_m(x) \quad (8)$$

$f(x)$ تابع ارزیابی x و w_1, w_2, \dots, w_m وزن های غیر منفی هستند که به صورت تصادفی تولید می شوند.

مرحله ۳: انتخاب: انتخاب عملگری است که دو والد را برای تولید یک فرزند جدید برمی گزیند. در انتخاب، یک احتمال انتخاب $P_s(x_i)$ برای هر عضو جمعیت بر حسب یک معیار اسکالر محاسبه می شود و با استفاده از roulette wheel دو والد مورد نظر مشخص می شوند.

$$P_s(x_i) = \frac{f(x_i) - f_{min}(x_\psi)}{\sum_{j=1}^N \{f(x_j) - f_{min}(x_\psi)\}} \quad (9)$$

$\forall i = 1, 2, \dots, N$

$f_{min}(x_\psi)$ کمترین مقدار fitness جمعیت نسل اخیر است. بر طبق این احتمال انتخاب، یک جفت از والدین از جمعیت استخراج می شوند.

مرحله ۴: باز ترکیبی؛ باز ترکیبی عملگری برای تولید یک فرزند جدید از والد، بر طبق احتمال انتخاب (P_c) عملگرهای باز ترکیبی متفاوتی برای GA ارائه شده است.

مرحله ۵: جهش؛ جهش عملگری برای تغییر اعضای داخل یک عضو جمعیت (رشته) که با کمک عمل باز ترکیبی ایجاد شده، می باشد. این عملگر موجب انتقال از جواب اخیر به جواب همسایه بر طبق احتمال انتخاب (P_m) می باشد. در روش ارائه شده، هر متغیری در رشته x_i در محدوده $[a_i, b_i]$ با احتمال (P_m) بر طبق رابطه زیر تغییر می کند.

$$x'_i = \begin{cases} x_i + \Delta(t, b_i - x_i) & \text{if } \delta = 0 \\ x_i + \Delta(t, x_i - a_i) & \text{if } \delta = 1 \end{cases} \quad (10)$$

$\Delta(t, y) = y[1 - r^{(1-t/t_{max})^\beta}]$

در رابطه فوق، r یک عدد تصادفی $[0, 1]$ ، t_{max} ، $r \in [0, 1]$ ماکزیمم تعداد generation است.

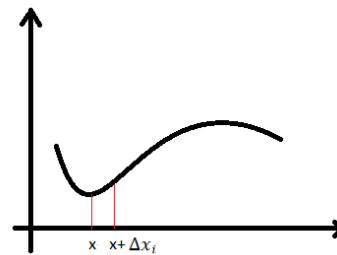
شایستگی در نقطه مطلق x_m بر حد مثبت و منفی نقطه x_m غلبه کند، مقدار x_m تغییری نخواهد کرد. در مرحله چهارم، اگر مقدار x_m تغییر نکرده باشد تا زمانی که هنوز به k_{max} نرسیده باشد، طول گام Δx_i بر طبق معادله $\Delta x_i = \Delta x_i(1 - r^{k/k_{max}})$ به روز می شود که در آن r یک عدد تصادفی است. $r \in [0,1]$

در مرحله پنجم، اگر x'_m بهتر باشد $f(x'_m) > f(x_m)$ ، نقطه جدید x'_m را تولید کرده و به مرحله ششم می رویم و در مرحله ششم با کمک نقطه های پایه x'_m و x_m یک جهت حرکت S به صورت $S = x'_m - x_m$ تعیین می کنیم. همچنین x''_m را به شکل $x''_m = x'_m + \lambda S$ می کنیم که در آن λ طول گام می باشد. در مرحله هفتم، اگر $f(x''_m) > f(x'_m)$ آنگاه $x_m = x'_m$ و $x'_m = x''_m$ را قرار داده و وارد مرحله ششم می شویم. در مرحله هشتم اگر $f(x'_m) \leq f(x''_m)$ ، آنگاه $x_m = x'_m$ را قرار داده و به مرحله چهارم می رویم. این مراحل بر روی مجموعه E^t اعمال می شود تا الگوریتم را قادر به جست و جوی نواحی کم جمعیت نماید. شبه کد الگوریتم ارائه شده در این مقاله همانند شکل (۸) خواهد بود. همان گونه که در شکل مشاهده می کنید، این الگوریتم به ترتیب شامل سه بخش الگوریتم کرم شب تاب چندهدفه، الگوریتم ژنتیک و جست و جوی محلی می باشد که در سه بخش ۱.۴، ۲.۴ و ۳.۴ به تفصیل توضیح داده شده است.

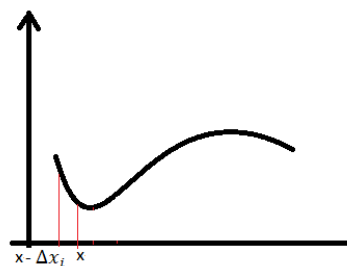
وجود فاصله بین جواب های غالب موجب ایجاد یک منحنی غیرهموار^۱ می شود. روش ارائه شده در این مقاله با الگوریتم های موجود مقایسه شده است و نشان داده می شود که الگوریتم ارائه شده موجب ایجاد پخش یکنواخت از نقاط در طول منحنی می شود و در نقاط انتهایی منحنی، نقاطی را می یابد. مشاهده می شود که الگوریتم ارائه شده یک توزیع یکنواخت از منحنی پرتو بهینه را ایجاد کرده و جواب های خیلی خوبی را در دو انتهای منحنی نشان می دهد در حالی که روش های قبل این توانایی را ندارند.

میزان شایستگی در نقطه $x - \Delta x_i$ و $x + \Delta x_i$ کمتر از نقطه x است، بنابراین بهترین نقطه همان نقطه x می باشد.

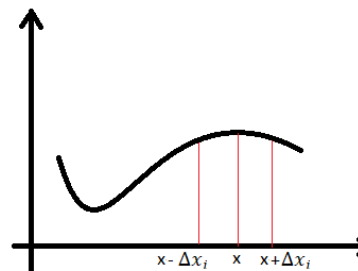
در نظر داشته باشید موارد بیان شده در این مثال یک شایستگی برای هر نقطه را نشان می دهد، در حالی که در مسائل بهینه سازی چندهدفه هر نقطه می تواند چند شایستگی متفاوت بر اساس هر هدف داشته باشد که در این صورت به جای رابطه بزرگ تری و کوچک تری در توضیحات فوق، باید از رابطه غلبه کردن استفاده شود.



شکل (۵): حالت اول رابطه (۱۱)



شکل (۶): حالت دوم رابطه (۱۱)



شکل (۷): حالت سوم رابطه (۱۱)

در این صورت، اگر میزان شایستگی در حد مثبت نقطه x_m بر نقطه مطلق x_m غلبه کند، نقطه x_m (یعنی همان کرم شب تاب m) را به اضافه Δx_i می کنیم؛ یعنی به سمت شایستگی بیشتر کمی متمایل می شویم. اگر میزان شایستگی در حد منفی نقطه x_m بر نقطه مطلق x_m و نیز حد مثبت x_m غلبه کند، از نقطه x_m مقدار Δx_i را کم می کنیم. همچنین اگر مقدار

1) Initialize parameters for Firefly and GA algorithms

2) Identify local set $L_i^{t=0} = \{X_i(t=0) \forall i = 1, \dots, N\}$

واکاوی بهتر جواب‌ها، نمودار و جداولی برای نتایج آزمایش‌ها ترسیم می‌کنیم. برای بررسی الگوریتم‌های مورد نظر، از دو مسئله محک در بهینه‌سازی چندهدفه استفاده شده است.

مسئله اول: به‌عنوان اولین مسئله از تابع محک استفاده شده که نسبت به دو مسئله بعدی از پیچیدگی کمتری برخوردار است. مشخصات این تابع به شرح زیر است:

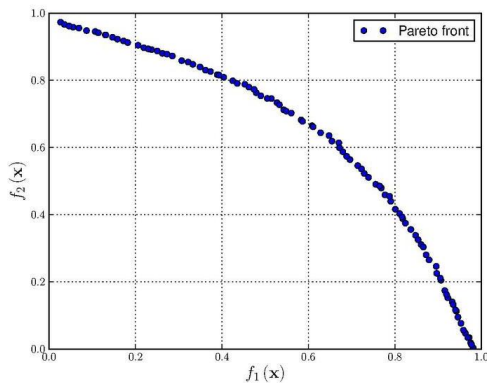
$$\text{Minimize } F = (f_1(\vec{x}), f_2(\vec{x})) \quad (12)$$

$$f_1(\vec{x}) = 1 - e^{-\sum_{i=1}^n (\frac{x_i}{\sqrt{n}})^2}$$

$$f_2(\vec{x}) = 1 - e^{-\sum_{i=1}^n (\frac{x_i+1}{\sqrt{n}})^2}$$

همچنین نمودار پرتوی بهینه برای این تابع در شکل (۹)

بیان شده است.



شکل (۹): نمودار پرتوی بهینه برای تابع Fonseca and Fleming

شکل‌های (۱۰) و (۱۱) نتایج حاصل از اجرای دو الگوریتم پایه (ترکیب الگوریتم PSO و الگوریتم ژنتیک) و الگوریتم پیشنهادی (ترکیب الگوریتم کرم شب تاب چندهدفه و الگوریتم ژنتیک) را نمایش می‌دهند. همان گونه که مشاهده می‌کنید در روش پایه که بر اساس الگوریتم PSO می‌باشد، ذرات بعد از ۱۰۰ تکرار هنوز پراکنده بوده و بر منحنی پرتو به‌طور کامل منطبق نشده‌اند. همچنین این ذرات به‌طور یکنواخت بر روی پرتوهای بهینه قرار نگرفته‌اند. درحالی‌که منحنی در روش ارائه شده که در شکل (۱۱) ارائه شده است جمعیت کرم شب تاب نهایی هموارتر بوده و کرم‌های شب تاب به‌صورت یکنواخت در طول منحنی پخش شده‌اند.

and $LP_i^{t=0} = \{X_i(t=0) \forall i = 1, \dots, N\}$

3) Identify global set $G^{t=0} = ND(\cup_{i=1}^N L_i^{t=0})$

4) The nearest member in $G^{t=0}$ to the i^{th} Firefly is $GP_i^{t=0}$

5) Set $E^{t=0} = G^{t=0}$

6) **while** Condition1 is not satisfied

// Multi-objective Firefly Algorithm

while Condition2 is not satisfied

for i=1 to ps **do**

for j=1 to ps **do**

/* movement through attraction*/

if ($f(x_j(t)) < f(x_i(t))$) **then**

Move $x_i(t)$ towards $x_j(t)$ according to Eq. 3;

Calculate the fitness value of $x_i(t)$ (new candidate solution);

end

end

end

t++;

end

// Multi-objective GA Algorithm

while Condition3 is not satisfied

Evaluate individual

Select best Parents

Do crossover and mutation

Repair

Select next generation

end

// Start Local Search

for a $X_m \in E^t$ generate X'_m

while ($f(X'_m) \neq f(X_m)$)

if $X'_m = X_m$ **then**

Reduce Δx_i and generate X'_m

end

Generate X''_m

if ($f(X''_m) > f(X'_m)$) **then**

$X_m = X'_m$ and $X'_m = X''_m$

Generate X''_m

else if ($f(X''_m) \neq f(X'_m)$) **then**

$X_m = X'_m$

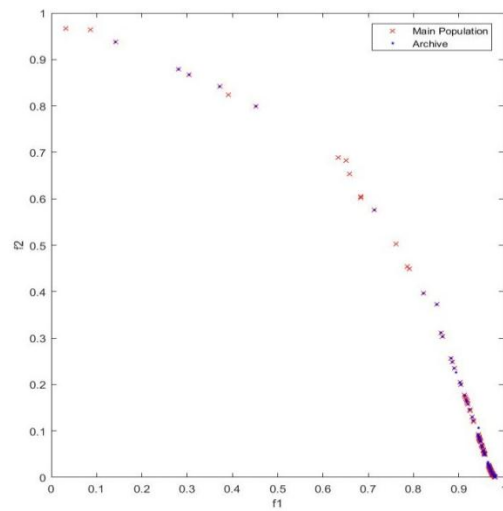
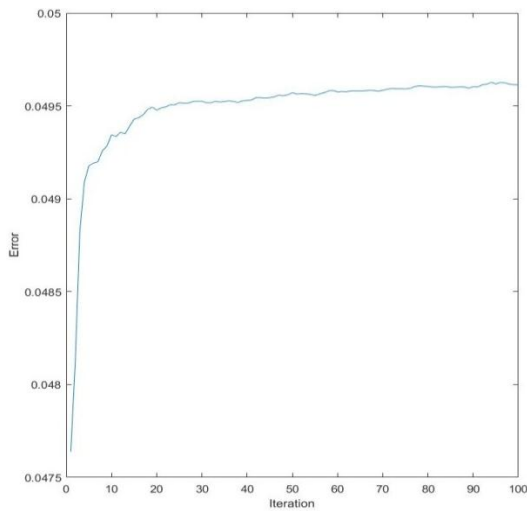
end

end

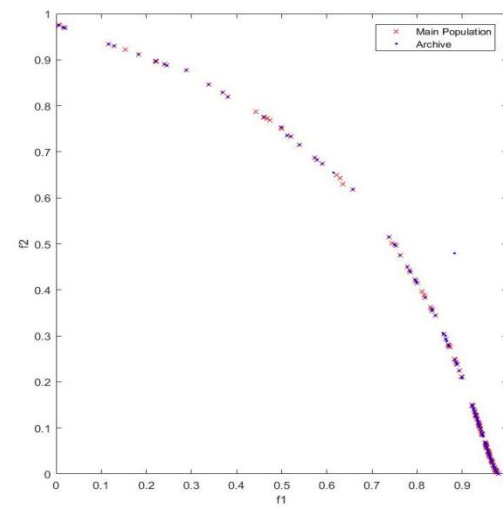
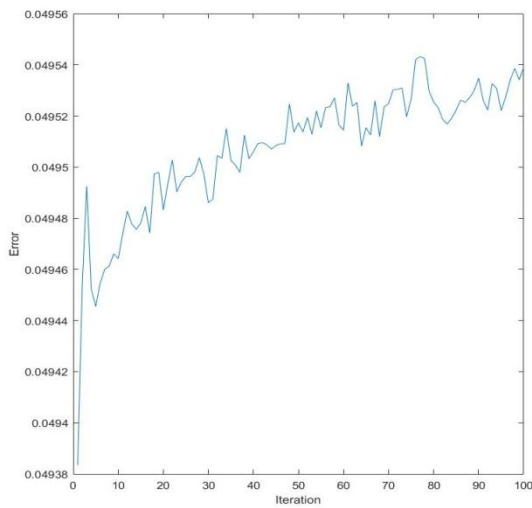
end

end

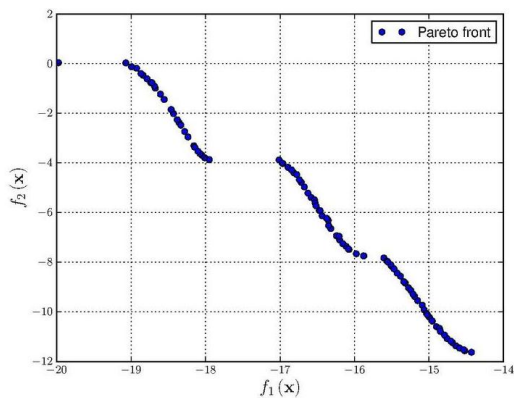
شکل (۸): شبه کد کلی الگوریتم ارائه شده در این مقاله بر اساس مقاله



شکل (۱۰): نحوه پراکندگی ۳۰۰ ذره حاصل از اجرای الگوریتم پایه در طول صد تکرار بر روی تابع رابطه (۱۲)



شکل (۱۱): نحوه پراکندگی ۳۰۰ کرم شب تاب حاصل از اجرای الگوریتم پیشنهادی در طول صد تکرار بر روی تابع رابطه (۱۲)



شکل (۱۲): نمودار پرتوی بهینه برای پایگاه داده kursawe

مسئله دوم: در این بخش از تابع kursawe به عنوان تابع

محک استفاده شده است. مشخصات این تابع به شرح زیر است:

$$\text{Minimize } F = (f_1(\vec{x}), f_2(\vec{x}))$$

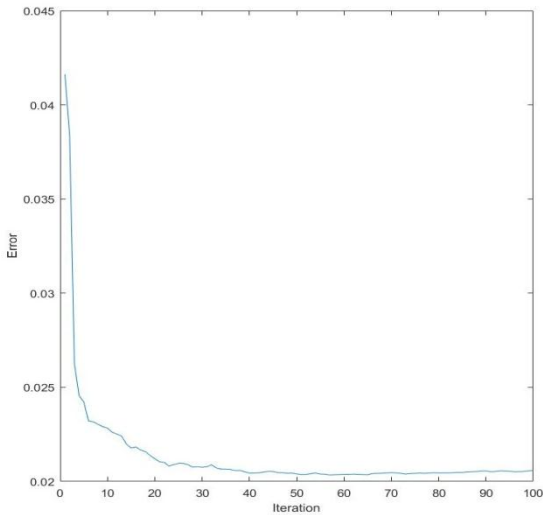
$$f_1(\vec{x}) = \sum_{i=1}^2 -10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} \quad (13)$$

$$f_2(\vec{x}) = \sum_{i=1}^2 (|x_i^{0.8} + 5\sin(x_i^3)|)$$

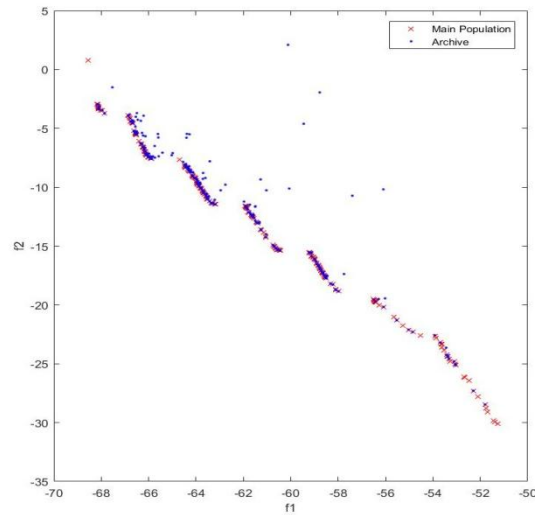
نمودار پرتوی بهینه برای این مجموعه داده در شکل (۱۲)

نشان داده شده است.

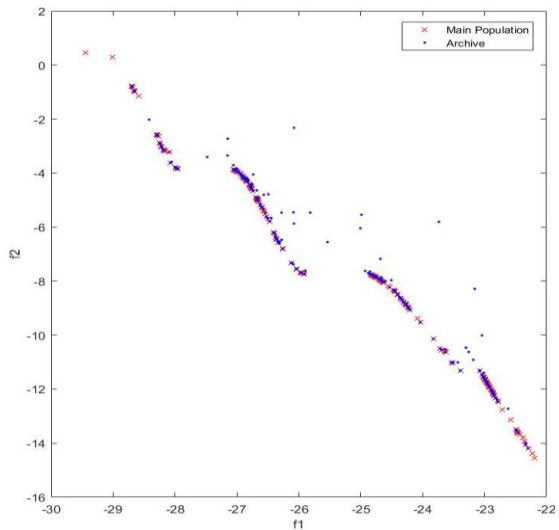
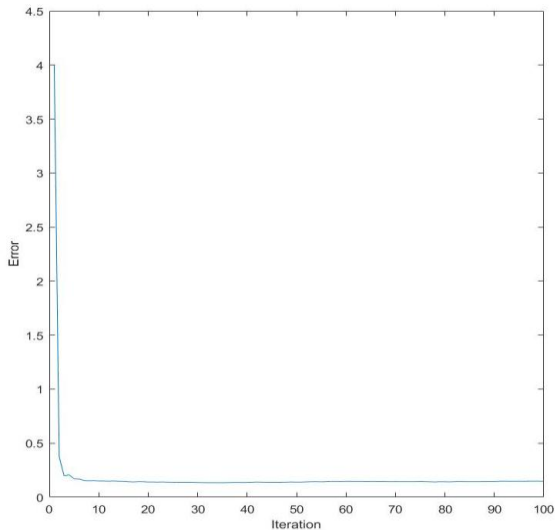
نشده‌اند. همچنین این ذرات به‌طور یکنواخت بر روی پرتوهای بهینه قرار نگرفته‌اند. درحالی‌که منحنی در روش ارائه شده که در شکل (۱۴) بیان شده، جمعیت کرم شب تاب نهایی اگرچه دارای پراکندگی است اما این پراکندگی به نسبت کمتر و هموارتر بوده و کرم‌های شب تاب به‌صورت یکنواخت‌تر در طول منحنی پخش شده‌اند.



شکل‌های (۱۳) و (۱۴) به ترتیب نتایج حاصل از اجرای دو الگوریتم پایه (ترکیب الگوریتم PSO و الگوریتم ژنتیک) و الگوریتم پیشنهادی (ترکیب الگوریتم کرم شب تاب چندهدفه و الگوریتم ژنتیک) بر روی تابع بیان شده در رابطه (۱۳) را نمایش می‌دهند. همان گونه که مشاهده می‌کنید، در روش پایه که بر اساس الگوریتم PSO می‌باشد، ذرات بعد از ۱۰۰ تکرار هنوز پراکنده بوده و بر منحنی پرتو به‌طور کامل منطبق



شکل (۱۳): نحوه پراکندگی ۳۰۰ ذره حاصل از اجرای الگوریتم پایه در طول صد تکرار بر روی تابع رابطه (۱۳)



شکل (۱۴): نحوه پراکندگی ۳۰۰ کرم شب تاب حاصل از اجرای الگوریتم پیشنهادی در طول صد تکرار بر روی تابع رابطه (۱۳)

برای بهینگی الگوریتم است که در مقاله مرجع هم از آن یاد شده و به‌عنوان معیار برتری الگوریتم نسبت به الگوریتم‌های ارائه شده پیشنهادی است.

همان طور که در شکل هم مشخص است، فاصله بین ذرات در شکل (۱۴) کمتر است و اعضا در این روش به‌طور هموار در طول منحنی پرتو جای گرفته‌اند؛ که این یک معیار

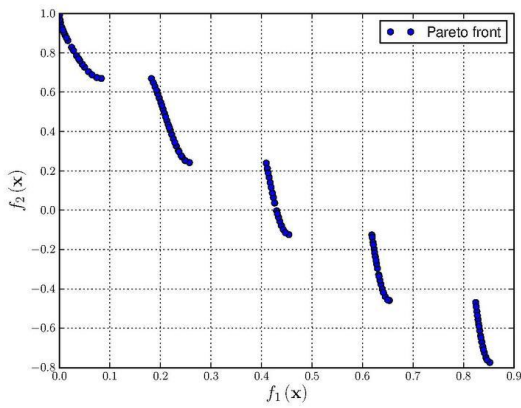
مسئله سوم: در این بخش از تابع Zitzler-Deb-Thiele

به عنوان محک استفاده شده است. مشخصات این تابع به شرح زیر است:

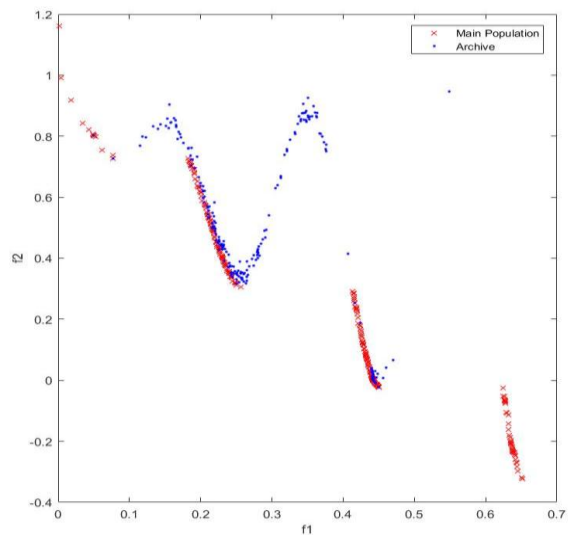
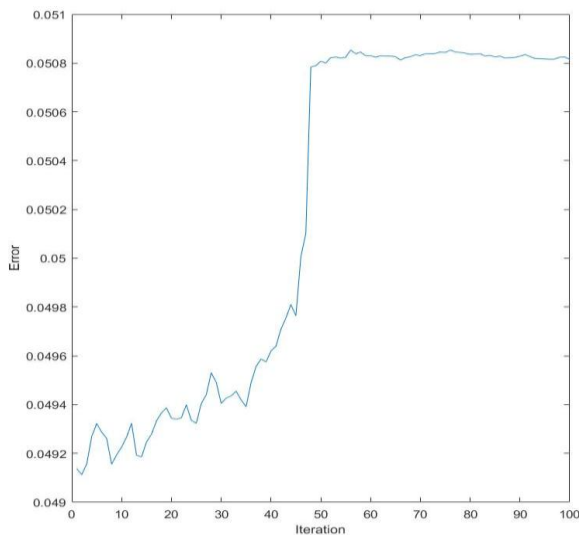
$$\begin{aligned}
 f_1(\vec{x}) &= x_1 \\
 f_2(\vec{x}) &= g(x)h(f_1(x), g(x)) \\
 g(x) &= 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \\
 h(f_1(x), g(x)) &= 1 - \sqrt{\frac{f_1(x)}{g(x)}} \\
 &\quad - \left(\frac{f_1(x)}{g(x)}\right) \sin(10\pi f_1(x))
 \end{aligned}
 \tag{۱۴}$$

همچنین نمودار پرتوی بهینه برای این تابع در شکل (۱۵)

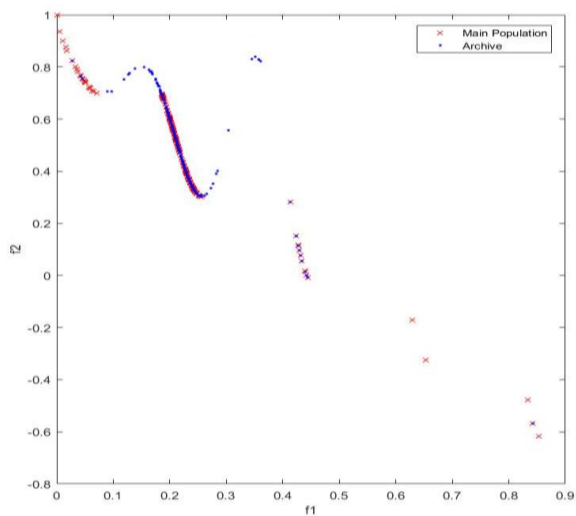
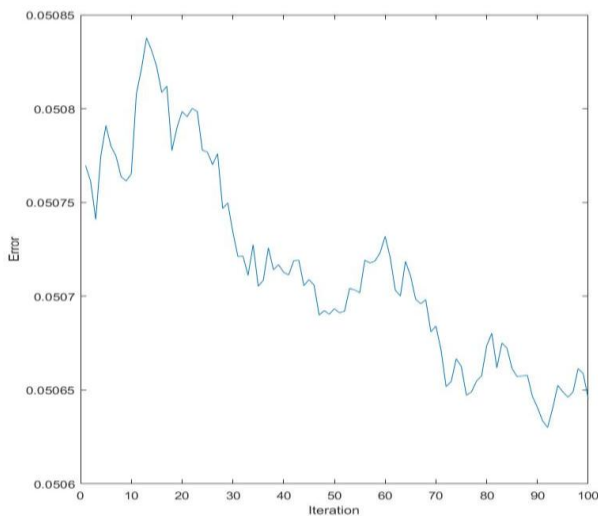
نمایش داده شده است.



شکل (۱۵): نمودار پرتوی بهینه برای پایگاه داده Zitzler-Deb-Thiele



شکل (۱۶): نحوه پراکندگی ۳۰۰ ذره حاصل از اجرای الگوریتم پایه در طول صد تکرار بر روی تابع رابطه (۱۴)



شکل (۱۷): نحوه پراکندگی ۳۰۰ کرم شب تاب حاصل از اجرای الگوریتم پیشنهادی در طول صد تکرار بر روی تابع رابطه (۱۴)

جدول (۱): مقایسه میزان خطای الگوریتم پایه و الگوریتم پیشنهادی

تعداد کرم های شب تاب (یا تعداد ذرات)	تعداد تکرار	فاصله پرتو- پرتو در روش پایه	فاصله پرتو- پرتو در روش پیشنهادی	فاصله ذره- پرتو در روش پایه	فاصله کرم شب تاب- پرتو در روش پیشنهادی
۱۰۰	۱۰۰	۰/۰۵۷۵	۰/۰۵۶۵	۰/۰۶۰۴	۰/۰۵۹۰
۲۰۰	۱۰۰	۰/۰۰۶	۰/۰۵۵۹	۰/۰۰۶۳	۰/۰۵۷۶
۳۰۰	۱۰۰	۰/۰۵۷۳	۰/۰۵۵۸	۰/۰۰۶۱۵	۰/۰۵۷۰

شده است که در آن به ترکیب الگوریتم PSO و الگوریتم ژنتیک بر اساس مشخصات بیان شده در این مقاله پرداخته است. تمامی آزمایش ها و مقایسه ها در این مقاله بین روش پیشنهادی شده و الگوریتم پایه ارائه شده است. همان گونه که در آزمایش انجام شده بر روی سه تابع محک گوناگون نشان داده شده، روش پیشنهادی موجب ایجاد منحنی های پرتو هموارتر و منطبق تر با پرتوهای بهینه نسبت به روش پایه شده است. همچنین با تغییراتی که در الگوریتم اعمال انجام شده، مشاهده کردیم که الگوریتم بهبود داده شده، باعث بهبود ویژگی یکنواختی منحنی پرتو می شود و فاصله بین نقاط موجود بر روی منحنی را کم و یکنواخت می کند و باعث کاهش خطای الگوریتم می شود. گفتنی است که الگوریتم ترکیبی کرم شب تاب و الگوریتم ژنتیک در مقالات دیگر نیز مورد استفاده قرار گرفته اما در آن کارها به صورت تک هدفه ارائه شده اند [۲۴ و ۲۵]، بنابراین قابل مقایسه با روش پیشنهادی در این مقاله نیستند.

محاسبه خطا: برای محاسبه خطا دو معیار در نظر گرفته شده است: ۱. فاصله پرتو- پرتو (Pareto-Pareto-Distance)؛ در این روش خطا، بر اساس میانگین فاصله میان نقاط پرتوی بهینه (حاصل از الگوریتم های اعمال شده) و پرتوی اصلی تابع محاسبه شده است. ۲. فاصله کرم شب تاب/ ذره- پرتو (Firefly/Particle-Pareto-Distance)؛ در این روش خطا بر اساس میانگین فاصله، میان ذرات جمعیت (حاصل از الگوریتم های اعمال شده) و پرتوی اصلی تابع محاسبه شده است. نتایج مربوط به خطای دو الگوریتم پایه و الگوریتم پیشنهادی در جدول (۱) بیان شده است. شایان ذکر است که این خطاها به صورت میانگین گزارش شده اند.

۶. نتیجه گیری

در این مقاله، یک الگوریتم چندهدفه ترکیبی جدید بر اساس الگوریتم کرم شب تاب چندهدفه و الگوریتم ژنتیک ارائه شده که با کمک الگوریتم ژنتیک و جست و جوی محلی بهبود یافته است. این الگوریتم بر اساس توسعه یک الگوریتم موجود ارائه

منابع

- [1] Coello, C.A.C., Lamont, G.B. and Van Veldhuizen, D.A., *Evolutionary algorithms for solving multi-objective problems*, Genetic and evolutionary computation series, ISBN 978-0-387-33254-3, Springer, 2007.
- [2] Miettinen, K., *Nonlinear multiobjective optimization*. International Series in Operations Research & Management Science, Vol. 12, ISSN 0884-8289, Springer Science & Business Media, 2012.
- [3] Trivedi, A., Srinivasan, D., Sanyal, K. and Ghosh, A., "A survey of multiobjective evolutionary algorithms based on decomposition", *IEEE Transactions on Evolutionary Computation*, Vol. 21, No. 3, pp. 440-462, 2016.
- [4] Reeves, C.R., "Genetic algorithms", in *Handbook of metaheuristics*: Vol. 146, pp. 109-139, Springer, 2010.
- [5] Yang, X.S., *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [6] Brest, J. and Maučec, M.S., "Self-adaptive differential evolution algorithm using population size reduction and three strategies", *Soft Computing*, Vol. 15, No. 11, pp. 2157-2174, 2011.
- [7] Tiwari, S., Fadel, G. Koch, P. and Deb, K., "Performance assessment of the hybrid archive-based micro genetic algorithm (AMGA) on the CEC09 test problems", in *IEEE Congress on Evolutionary Computation*, pp. 1935-1942, 2009.
- [8] Qu, B.Y. and Suganthan, P.N., "Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster", in *IEEE Congress on Evolutionary Computation*, pp. 2934-2939, 2009.
- [9] Ali, M., Siarry, P. and Pant, M., "An efficient differential evolution based algorithm for solving multi-objective optimization problems", *European journal of operational research*, Vol. 217, No. 2, pp. 404-416, 2012.
- [10] Noghianian, S., Sabouni, A., Desell, T. and Ashtari, A., "Global optimization: Differential evolution, genetic algorithms, particle swarm, and hybrid methods", in *Microwave Tomography*: Springer, pp. 39-61, 2014.
- [11] Mahmoodabadi, M.J., Safaie, A.A., Bagheri, A. and

- Nariman-Zadeh, N., "A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model", *Applied Soft Computing*, Vol. 13, No. 5, pp. 2577-2591, 2013.
- [12] Yang, X.S., "Firefly algorithm", *Nature-inspired metaheuristic algorithms*, Vol. 20, pp. 79-90, 2008.
- [13] Yang, X.S., *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.
- [14] Yang, X.S., "Firefly algorithms for multimodal optimization", in *International symposium on stochastic algorithms*, pp. 169-178: Springer, 2009.
- [15] Hornig, M.H. and Jiang, T.W., "The codebook design of image vector quantization based on the firefly algorithm", in *International Conference on Computational Collective Intelligence*, pp. 438-447: Springer, 2010.
- [16] Apostolopoulos, T. and Vlachos, A., "Application of the firefly algorithm for solving the economic emissions load dispatch problem", *International journal of combinatorics*, Vol. 2011, 2010.
- [17] Yang, X.S., "Multiobjective firefly algorithm for continuous optimization", *Engineering with computers*, Vol. 29, No. 2, pp. 175-184, 2013.
- [18] Wang, H. and et al., "A hybrid multi-objective firefly algorithm for big data optimization", *Applied Soft Computing*, Vol. 69, pp. 806-815, 2018.
- [19] Amiri, B., Hossain, L., Crawford, J.W. and Wigand, R.T., "Community detection in complex networks: Multi-objective enhanced firefly algorithm", *Knowledge-Based Systems*, Vol. 46, pp. 1-11, 2013.
- [20] Mühlenbein, H., "Genetic algorithms", In *Local Search in Combinatorial Optimization*, Edited by: Aarts, E. and Lenstra, J. K., John Wiley & Sons, New York, pp. 137-172, 1997.
- [21] A .Mousa, M. El-Shorbagy, and W. F. Abd-El-Wahed, "Local search based hybrid particle swarm optimization algorithm for multiobjective optimization," *Swarm and Evolutionary Computation*, Vol. 3, pp. 1-14, 2012.
- [22] Pal, S.K., Rai, C. and Singh, A.P., "Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems", *International Journal of intelligent systems and applications*, Vol. 4, No. 10, p. 50, 2012.
- [23] Lohrer, M., "A comparison between the firefly algorithm and particle swarm optimization," *Computer Science*, 2013.
- [24] Wahid, F., Ghazali, R. and Ismail LH., "Improved firefly algorithm based on genetic algorithm operators for energy efficiency in smart buildings", *Arabian Journal for Science and Engineering*. Vol. 44, No. 4, pp.4027-47, 2019.
- [25] Maheshwar, K.K. and Arora, V., "A hybrid data clustering using firefly algorithm based improved genetic algorithm", *Procedia Computer Science*, Vol. 58, pp.249-256, 2015.