

## یک الگوریتم بهینه‌سازی بیز مبتنی بر خوشه‌بندی وقتی برای محیط‌های پیوسته

بهناز مرادآبادی حمید بیگی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

### چکیده

الگوریتم‌های تخمین توزیع<sup>۱</sup> دسته‌ای از الگوریتم‌های تکاملی<sup>۲</sup> هستند که از توزیع آماری جمعیت برای تولید نسل بعد استفاده می‌کنند. الگوریتم بهینه‌سازی بیز<sup>۳</sup>، یکی از الگوریتم‌های تخمین توزیع است که با تولید یک شبکه بیز تلاش می‌کند مدل آماری جمعیت را محاسبه نماید و فرزندان را از روی مدل آماری ساخته شده تولید کند. در این مقاله یک الگوریتم بهینه‌سازی بیز برای حل مسایل بهینه‌سازی پیوسته ارائه می‌شود. این الگوریتم با استفاده از روش‌های خوشه‌بندی وقتی، افراد جمعیت را خوشه‌بندی نموده و سپس برای هر خوشه، یک شبکه بیز تولید می‌کند. استفاده از روش‌های خوشه‌بندی وقتی، سبب جستجوی بهتر فضا در ابتدای الگوریتم و جستجوی دقیق‌تر فضا در انتهای الگوریتم می‌شود. نتایج پیاده‌سازی‌ها حاکی از برتری الگوریتم پیشنهادی با استفاده از روش‌های خوشه‌بندی فازی وقتی و خوشه‌بندی  $k'$ -means وقتی در مقایسه با الگوریتم بهینه‌سازی بیز استاندارد است.

**کلمات کلیدی:** الگوریتم‌های تکاملی، شبکه‌های بیز، الگوریتم‌های تخمین توزیع.

### ۱- مقدمه

ژنوم یا کروموزوم می‌نامند. شبه کد این الگوریتم در شکل ۱ نمایش داده شده است.

۱. تولید جمعیتی از راه‌حل‌های اولیه.
۲. محاسبه تابع برازش برای راه‌حل‌ها.
۳. انتخاب تعدادی راه‌حل به عنوان والدین برای تولید نسل بعد.
۴. انجام عملیات ترکیب با استفاده از والدین انتخاب‌شده.
۵. انجام عملیات جهش بر روی فرزندان تولیدشده.
۶. ادغام دو نسل بر اساس روش‌های جایگزینی.
۷. خاتمه الگوریتم در صورت رسیدن به شرط خاتمه و در غیر این صورت رفتن به مرحله ۳.

شکل ۱- شبه کد الگوریتم ژنتیک

نیاز به جستجو در حل مسایل کاربردی امری غیر قابل اجتناب و درعین حال دشوار است. به همین جهت تعداد زیادی از الگوریتم‌های جستجو با فلسفه‌ها و دامنه‌های استفاده متفاوت بوجود آمده‌اند. الگوریتم‌های تکاملی از تکامل در طبیعت نشأت گرفته و از روش‌های بهینه‌سازی تصادفی<sup>۴</sup> هستند. این الگوریتم‌ها، الگوریتم‌های جستجوی تصادفی هستند که تکامل زیست‌شناسی را تقلید می‌کنند. آنها بر جمعیتی از پاسخ‌های ممکن مساله با در نظر گرفتن اصل زنده ماندن بهترین اعضا، اعمال می‌شوند تا پاسخ‌های بهتر و بهتر مساله را با توجه به تابع برازش مساله تولید کنند. قدرتمندی الگوریتم‌های تکاملی در بسیاری از زمینه‌های کاربردی سبب تولید الگوریتم‌های متعددی برای بهبود کارایی آنها شده است. در این میان الگوریتم‌های ژنتیک<sup>۵</sup> دسته بزرگی از این الگوریتم‌ها را تشکیل می‌دهند [۱].

در الگوریتم‌های ژنتیک، نخست مجموعه‌ای از راه‌حل‌ها انتخاب شده و سپس نسل بعد با استفاده از عملگرهای ترکیب<sup>۶</sup> و جهش<sup>۷</sup> تولید می‌شوند. مجموعه جواب‌های تولید شده در هر دوره الگوریتم را یک نسل و هر کدام از جواب‌ها را یک

هر ژنوم ترکیبی از متغیرهای مساله است. در برخی مسایل، متغیرهای مساله به هم وابسته و در برخی دیگر از هم مستقل هستند؛ ولی همواره ارتباط و تبادل اطلاعات بین ژنوم‌ها از طریق انتخاب و ترکیب مجدد ژنوم‌ها در یک نسل صورت

در ژنوم قرار گرفته باشند، کارایی مناسبی دارند. چرا که اگر بین متغیرهای یک مساله روابطی منطقی وجود داشته باشد، استفاده از عملگرهای ساده و ثابت جهش و ترکیب سبب از بین رفتن روابط بین متغیرها در نسل بعد می‌شود. به همین دلیل گاهی در ترکیب ژنوم‌ها با یکدیگر نه تنها بهبودی در کیفیت جواب‌ها حاصل نمی‌شود، بلکه الگوریتم در نقاط بهینه محلی به دام می‌افتد. در مراجع [۲، ۳]. دسته‌ای از الگوریتم‌های مکاشفای غیرقطعی مبتنی بر جمعیت با نام الگوریتم‌های تخمین توزیع ارائه شده‌اند که همانند الگوریتم‌های ژنتیکی نیازمند فضای جستجوی هموار و معادلات دیفرانسیل پیچیده نمی‌باشند و علاوه بر این بسیاری از مشکلات الگوریتم‌های ژنتیکی را حل نموده‌اند. در الگوریتم‌های تخمین توزیع، با ساخت یک مدل احتمالاتی از اجزای سازنده ژنوم، سرعت پیشروی به سوی جواب بهینه مساله افزایش می‌یابد. در این الگوریتم‌ها، جمعیت جدید با استفاده از عملگرهای ترکیب و جهش به وجود نمی‌آیند؛ بلکه ژنوم‌های جدید، بر اساس توزیع احتمالاتی تخمین زده شده بر اساس ژنوم‌های انتخاب شده از نسل‌های قبل، نمونه برداری و ساخته می‌شوند. در واقع تخمین همین تابع توزیع، مشکل‌ترین بخش الگوریتم‌های تخمین توزیع است. قالب کلی الگوریتم‌های تخمین توزیع به صورت زیر است:

گام ۱- جمعیت نخستین،  $D_0$ ، با تعداد  $N$  ژنوم تشکیل می‌شود. معمولاً تولید  $N$  ژنوم مذکور، با فرض توزیع یکنواخت به روی هر یک از متغیرها صورت می‌پذیرد.

سپس هر یک از ژنوم‌ها با کمک تابع برازش ارزیابی می‌شوند.

گام ۲- در نسل  $i$ ام، یک تعداد مشخص  $Se$  ( $Se \leq N$ ) ژنوم از میان جمعیت کنونی،  $(D_i)$ ، بر اساس یک معیار تعریف شده (معمولاً بهترین‌ها به بر اساس تابع برازش مساله موردنظر به صورت قطعی و یا احتمالی) انتخاب می‌شوند که با  $D_i^{se}$  مشخص می‌شود.

گام ۳- یک مدل احتمالاتی  $n$  بعدی که نشان‌دهنده توزیع  $n$  متغیر موجود در راه حل می‌باشد، تخمین زده می‌شود.

گام ۴- در پایان یک جمعیت جدید متشکل از  $N$  ژنوم جدید با استفاده از توزیع احتمالاتی به دست آمده در گام ۳، ایجاد می‌شود.

گام ۵- در صورت برقرار نبودن شرط خاتمه الگوریتم به گام ۲ رفته و از آنجا ادامه پیدا می‌کند. در غیر این صورت بهترین ژنوم جمعیت به عنوان جواب نهایی اعلام می‌شود.

شرایط خاتمه متفاوتی مانند تولید تعداد معینی نسل، انجام تعداد معینی ارزیابی، یکنواختی در چندین نسل پایانی و ثابت ماندن تقریبی ارزش ژنوم‌ها در چند نسل پایایی را می‌توان در نظر گرفت [۲].

فرض کنید  $n$ ،  $i=1, \dots, n$  یک متغیر تصادفی و  $x_i$  یک مقدار برای  $X_i$  باشد، در این صورت  $X=(X_1, \dots, X_n)$  یک متغیر تصادفی  $n$  بعدی است که می‌تواند  $x=(x_1, \dots, x_n)$  را به عنوان مقدار خود بپذیرد. مدل گرافیکی دارای دو جزء تشکیل دهنده ساختار  $\zeta$  و مجموعه‌ای از چگالی‌های احتمالاتی محلی است؛ به طوری که ساختار  $\zeta$  به صورت یک گراف جهت‌دار بدون دور تعریف می‌شود. این گراف نشان دهنده ساختار وابستگی بین متغیرها است [۱۴]. اگر در این ساختار،  $Pa_i$  به معنای مجموعه متغیرهای والد متغیر  $X_i$  باشد، آنگاه متغیر  $X_i$  از تمام متغیرهای  $Pa_i - \{X_1, \dots, X_n\}$  مستقل بوده و بنابراین می‌توانیم تابع چگالی عمومی توام  $\sigma(X=x)$  را به صورت زیر بنویسیم:

$$\sigma(X=x) = \prod_{i=1}^n \sigma(x_i | Pa_i) \quad (1)$$

از طرفی با فرض این که چگالی‌های احتمالاتی محلی به مجموعه پارامترهای  $\theta_\zeta \in \Theta_\zeta$  که  $\theta_\zeta = \{\theta_1, \dots, \theta_n\}$  می‌باشد، وابسته است، می‌توانیم معادله (۱) را به صورت زیر بنویسیم:

می‌پذیرد. این جابجایی اطلاعات سبب می‌شود تا جواب‌های جزئی<sup>۱</sup> با یکدیگر ترکیب و احتمالاً جواب‌هایی با کیفیت بالاتر بدست آیند. اما با تمام ویژگی‌های مثبتی که در الگوریتم ژنتیک استاندارد وجود دارد، این الگوریتم تنها در مواقعی که متغیرها از هم مستقل و یا در فاصله کمی از هم در ژنوم قرار گرفته باشند، کارایی مناسبی دارد. چرا که اگر بین متغیرهای یک مساله روابطی منطقی وجود داشته باشد، استفاده از عملگرهای ساده و ثابت جهش و ترکیب سبب از بین رفتن روابط بین متغیرها در نسل بعد می‌شود. به عبارت دیگر رفتار الگوریتم ژنتیکی وابستگی شدیدی به پارامترهایی از جمله نمایش ژنوم، عملگرهای جهش و تولید نسل، احتمال جهش، اندازه جمعیت و تعداد نسل‌های تولید شده دارد. به همین دلیل گاهی در ترکیب ژنوم‌ها با یکدیگر نه تنها بهبودی در کیفیت جواب‌ها حاصل نمی‌شود، بلکه الگوریتم در نقاط بهینه محلی به دام می‌افتد [۱]. از طرفی یافتن روابط منطقی بین متغیرهای یک مساله، خود یک مساله تمام سخت<sup>۲</sup> است که با نام مساله پیوند<sup>۱</sup> شناخته می‌شود و با توجه به موارد بیان‌شده، الگوریتم‌های ژنتیک قادر به یافتن روابط منطقی میان متغیرها و حل مساله پیوند نمی‌باشند [۳]. یکی از راه‌های بهبود این مشکل، استفاده از الگوریتم‌های تخمین توزیع است که نخست توزیع احتمال حاکم بر راه‌حل‌های مساله را تخمین می‌زنند و سپس نسل بعد را بر اساس احتمال تخمین زده شده تولید می‌نمایند [۲، ۳]. این دسته از الگوریتم‌ها در واقع تلاش می‌کنند از عملگرهای ترکیب و جهش که باعث از بین رفتن روابط بین متغیرها می‌شوند، استفاده نکرده و مدل‌های احتمالی و توزیع‌های احتمالی را برای تولید نسل بعد به کار گیرند. الگوریتم بهینه‌سازی بیز نیز یکی از اعضای خانواده الگوریتم‌های تخمین توزیع است که برای تخمین توزیع احتمال موجود، یک شبکه بیز را که معرف روابط منطقی میان متغیرهای مساله است، ساخته و سپس نسل بعد را با استفاده از شبکه ساخته شده تولید می‌کند [۴، ۵].

در این مقاله، یک الگوریتم بیز بهبود یافته برای محیط‌های پیوسته ارائه می‌گردد. مشکل عمده الگوریتم بهینه‌سازی بیز ساخت یک شبکه بیز برای تمامی راه‌حل‌های منتخب می‌باشد. این عمل سبب می‌شود تا شبکه ساخته شده کیفیت لازم را نداشته باشد. برای رفع این مشکل، در این مقاله یک الگوریتم بهینه‌سازی بیز ارائه می‌شود که در آن نخست راه‌حل‌های منتخب با استفاده از الگوریتم‌های خوشه‌بندی وفقی<sup>۱۱</sup> به تعدادی خوشه افزای شده و سپس برای هر خوشه یک شبکه بیز ساخته می‌شود. از آنجا که در این‌گونه خوشه‌بندی نیازی به دانستن تعداد دقیق خوشه‌ها قبل از خوشه‌بندی نمی‌باشد و تعداد مناسب خوشه‌ها در طول خوشه‌بندی تعیین خواهد شد، در ابتدای الگوریتم و با توجه به بالا بودن تنوع جمعیت، تعداد خوشه‌های تولید شده بیشتر است که سبب جستجوی بهتر فضا می‌شود و در انتهای الگوریتم و با توجه به پایین بودن تنوع جمعیت، تعداد خوشه‌های تولید شده کمتر و جستجو دقیق‌تر می‌شود. نتایج پیاده‌سازی‌ها نیز نمایش‌دهنده برتری الگوریتم پیشنهادی بر الگوریتم بهینه‌سازی بیز استاندارد است.

ادامه مقاله به صورت زیر سازماندهی شده است: بخش ۲، الگوریتم‌های تخمین توزیع را شرح می‌دهد. در بخش ۳، الگوریتم‌های تخمین توزیع برای محیط‌های پیوسته به صورت مختصر شرح داده می‌شود. شرح مختصری از الگوریتم‌های خوشه‌بندی وفقی در بخش ۴ آورده شده است. الگوریتم پیشنهادی بهینه‌سازی بیز در محیط‌های پیوسته در بخش ۵ ارائه می‌شود. نتایج تجربی حاصل از الگوریتم پیشنهادی، در بخش ۶ و نتیجه‌گیری نیز در بخش ۷ آورده خواهد شد.

## ۲- الگوریتم‌های تخمین توزیع

با تمام ویژگی‌های مثبتی که الگوریتم‌های تکاملی مانند الگوریتم‌های ژنتیک دارند، این الگوریتم‌ها تنها در مواقعی که متغیرها از هم مستقل و یا در فاصله کمی از هم

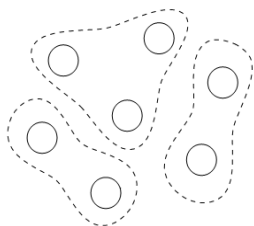
الگوریتم COMIT روابط دوتایی میان متغیرها را به صورت یک درخت نمایش می‌دهد و این الگوریتم تلاش در یافتن درختی از ژن‌ها دارد که روابط میان متغیرها را به درستی بازنمایی کند.

در الگوریتم BMDA روابط دوتایی میان متغیرها به صورت یک جنگل نمایش داده می‌شود و الگوریتم تلاش در یافتن جنگلی از ژن‌ها دارد که روابط میان متغیرها را به درستی نمایش دهد.

مدل سوم که در واقع عمومی‌ترین مدل است، قادر به تعیین روابط و وابستگی‌های چندتایی در میان متغیرها می‌باشد. چند نکته در این مدل حائز اهمیت است. از آنجا که این مدل احتمالاتی پیچیده است، تخمین آن کاری زمان‌بر است. بنابراین باید به گونه‌ی کارایی فرایند استخراج مدل بهبود یابد. از طرفی باید آستانه مناسبی برای رهایی از ارتباطات غیرضروری بین متغیرها، به کار گرفته شود. از جمله الگوریتم‌هایی که در این گروه جای می‌گیرند، می‌توان به FDA<sup>۱۲</sup>، BOA<sup>۱۳</sup> و ECGA<sup>۱۴</sup> اشاره کرد.

الگوریتم FDA، الگوریتمی است که تلاش می‌کند تنها و تنها احتمالات شرطی حاکم بر متغیرها را بیابد و نه ساختار شبکه‌ای حاکم بر آن را. بدین صورت که ابتدا مساله را تجزیه کرده و سپس تلاش می‌کند با استفاده از معیارهای آماری، احتمال حاکم بر قسمت‌های تجزیه شده را بیابد [۱۲].

در الگوریتم ECGA، نیز ابتدا متغیرهای موجود در مساله، به چند دسته افزاز شده و سپس با هر زیرمجموعه به صورت یک مدل احتمالی با توزیع یکنواخت رفتار می‌شود که این توزیع یکنواخت از داده‌های مربوط به هر خوشه محاسبه می‌شود [۱۳]. شکل ۲ نمای کلی برخورد ECGA را با متغیرهای یک مساله نمایش می‌دهد.



شکل ۲- نمای کلی برخورد ECGA با متغیرهای مساله

الگوریتم بهینه‌سازی بیز نیز که در حال حاضر قدرتمندترین الگوریتم موجود برای مدل کردن وابستگی‌های موجود در یک مساله بهینه‌سازی است، تلاش می‌کند یک شبکه بیز را برای نمایش وابستگی‌های موجود ساخته و نسل بعد را بر اساس شبکه ساخته‌شده تولید نماید. اگرچه این الگوریتم قادر به یافتن روابط میان متغیرها می‌باشد، اما ساختن یک شبکه بیز که به صورت بهینه روابط میان متغیرها را بازگو نماید، یک مساله تمام سخت است [۲۸]. بدین منظور برای ساخت این شبکه، معمولاً از الگوریتم‌های حریمانه استفاده می‌شود. یادآوری می‌شود که شبکه بیز در الگوریتم‌های بهینه‌سازی بیز عبارت است از گرافی جهت‌دار و بدون دور که در آن برای هر بعد مساله یک راس وجود دارد و همچنین یال‌های موجود در آن نیز نمایشگر وابستگی‌های موجود در میان ابعاد مساله است [۵]. نمایی از یک شبکه بیز در شکل ۳ نمایش داده شده است.

در این الگوریتم ابتدا یک مجموعه تصادفی از راه‌حل‌ها تولید می‌شود، سپس راه‌حل‌ها با استفاده از یک تابع برازش ارزیابی و مرتب شده، بخشی از آن‌ها به عنوان راه‌حل‌های امیدبخش برای تولید نسل بعد به صورت قطعی انتخاب می‌گردند؛ در نهایت یک شبکه بیز برای تمامی راه‌حل‌های منتخب تولید می‌شود. این شبکه شامل یک و یا تعدادی درخت می‌باشد که نمایانگر بلوک‌های سازنده موجود در آن مساله است. سپس، الگوریتم این درخت‌ها را از شبکه بیز استخراج نموده و برای هر درخت (بلوک سازنده) یک مدل احتمالی شرطی بر اساس ابعاد

$$\sigma(X = x | \theta_{\zeta}) = \prod_{i=1}^n \sigma(x_i | Pa_i, \theta_{\zeta}) \quad (2)$$

بنابراین مدل گرافیکی احتمالاتی را می‌توان به صورت  $M(\zeta, \theta_{\zeta})$  بیان کرد [۳]. بر اساس نوع ارتباطات بین متغیرها، مدل‌های گرافیکی احتمالاتی به سه گروه مستقل، با ارتباطات دوتایی و با ارتباطات چندتایی تقسیم می‌شوند. در گروه نخست، متغیرها از هم مستقل بوده و بنابراین احتمال یک نمونه  $x$  برابر حاصلضرب احتمال متغیرهای آن است:

$$\sigma(X = x) = \prod_{i=1}^n \sigma(x_i) \quad (3)$$

این مدل، ساده‌ترین مدل ممکن است و با قبول درصدی از خطا می‌توان از آن برای حل مسایل بهینه‌سازی غیرخطی استفاده کرد. از الگوریتم‌های این دسته می‌توان به UMDA<sup>۱۲</sup> [6]، PBIL<sup>۱۳</sup> [7] و CGA<sup>۱۴</sup> [۸] اشاره کرد که فقط برای مسایلی با متغیرهای دودویی که تنها مقادیر صفر و یک را می‌پذیرند، به کار می‌روند.

الگوریتم UMDA، مجموعه‌ای از ژنوم‌ها را نگهداری نموده و از آنها برداری برای نمایش احتمالات حاکم بر ژنوم‌ها می‌سازد و سپس نسل بعد را از بردار حاصل تولید می‌کند. الگوریتم PBIL، تلاش می‌کند به جای نگهداری جمعیت ژنوم‌ها، فقط یک بردار احتمال که احتمال ۱ بودن هر ژن را نمایش می‌دهد، ذخیره کند و در هر مرحله تعدادی ژنوم را از بردار موجود تولید کند. سپس بردار احتمال را برای هر ژن بر حسب  $X$  که بهترین فرد تولید شده است به صورت زیر به روزرسانی می‌کند:

$$p_i = (1 - \gamma) p_i + \gamma x_i \quad (4)$$

که  $0 \leq \gamma \leq 1$  ثابتی برای تعیین گام‌های به روزرسانی بردار احتمال است. الگوریتم CGA، نیز از روشی مشابه PBIL استفاده می‌کند. با این تفاوت که بعد از تولید نسل جدید از بردار احتمال موجود با فرض اینکه  $X$  بهترین فرد و  $y$  بدترین فرد باشد، بردار احتمال برای ۱ بودن هر ژن را به صورت زیر به روزرسانی می‌کند:

$$p_i = \begin{cases} p_i + \frac{1}{N} & \text{if } x_i = 1 \text{ and } y_i = 0 \\ p_i - \frac{1}{N} & \text{if } x_i = 0 \text{ and } y_i = 1 \\ p_i & \text{if } x_i = y_i \end{cases} \quad (5)$$

نادیده‌گرفتن وابستگی بین متغیرها، در بسیاری از مسایل بهینه‌سازی دور از واقعیت است. با در نظر گرفتن وابستگی بین متغیرها، اولین و ساده‌ترین نوع ارتباطات، ارتباطات دوتایی است. الگوریتم‌هایی که در این دسته قرار می‌گیرند، اغلب با استفاده از الگوریتم‌های حریمانه<sup>۱۵</sup>، مدل‌های احتمالاتی محلی را استخراج می‌کنند. از الگوریتم‌های این گروه می‌توان به MIMIC<sup>۱۶</sup> [۹]، COMIT<sup>۱۷</sup> [۱۰]، BMDA<sup>۱۸</sup> [۱۱] اشاره کرد.

در الگوریتم MIMIC روابط دوتایی میان متغیرها به صورت یک زنجیره نمایش داده می‌شود و این الگوریتم تلاش در یافتن زنجیره‌ای از ژن‌ها دارد که روابط میان متغیرها را به درستی بازنمایی کند.

توجه به استفاده این الگوریتم از مدل ترکیب گوسی، این الگوریتم می‌تواند برای بهینه‌سازی مسایل چندقله‌ای به کار رود، اما وجود یک ساختار برای نمایش وابستگی تمام اجزای مدل ترکیب گوسی سبب عدم کارایی در توابعی می‌شود که دارای وابستگی‌های متفاوت در قله‌های متفاوت هستند [۲۷].

در مراجع [۱۶، ۱۵]، یک الگوریتم بهینه‌سازی بیز برای بهینه‌سازی مسایل در محیط‌های پیوسته، با استفاده از مدل ترکیب گوسی ارائه شده است. شبه کد این الگوریتم در شکل ۴ نمایش داده شده است.

۱.	تولید جمعیتی از راه‌حل‌های اولیه.
۲.	محاسبه تابع برازش برای راه‌حل‌ها.
۳.	انتخاب تعدادی از راه‌حل‌ها برای تولید شبکه بیز.
۴.	ساخت یک شبکه بیز از راه‌حل‌های منتخب.
۵.	تولید نسل بعد با استفاده از شبکه بیز.
۶.	ادغام دو نسل بر اساس شایسته‌سالاری.
۷.	خاتمه الگوریتم در صورت رسیدن به شرط خاتمه و در غیر این صورت رفتن به گام ۳.

شکل ۴- شبه کد الگوریتم بهینه‌سازی بیز در محیط‌های پیوسته

شبکه بیز در این الگوریتم، با استفاده از معیار اطلاعاتی بیز که در ادامه توضیح داده خواهد شد، ساخته می‌شود. در نهایت این شبکه شامل یک و یا تعدادی درخت می‌باشد که نمایانگر بلوک‌های سازنده موجود در آن مساله است. الگوریتم، سپس این درخت‌ها را از شبکه بیز استخراج نموده و برای هر درخت (بلوک سازنده) یک مدل ترکیب گوسی بر اساس ابعاد مربوط به آن بلوک سازنده و با استفاده از راه‌حل‌های منتخب می‌سازد. سپس داده‌های مربوط به هر بلوک سازنده برای نسل بعد، با تولید اعداد تصادفی از مدل ترکیب گوسی مربوط به آن بلوک سازنده تولید می‌شود.

این بلوک‌های سازنده در کنار یکدیگر قرار گرفته و نسل بعد را تولید می‌کنند. این مراحل تا آنجا که شرط توقف الگوریتم ارضا شود، تکرار می‌پذیرد. مجدداً لازم به ذکر است که شرط توقف می‌تواند رسیدن به حداکثر تعداد اجرا و یا رسیدن به یک جواب قابل قبول باشد.

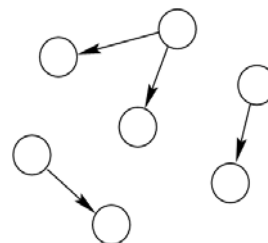
یکی از مشکلات این الگوریتم، استفاده و ساخت یک شبکه بیز برای مدل‌سازی تمامی راه‌حل‌های منتخب است؛ اگرچه استفاده از مدل ترکیب گوسی سبب جستجوی بهتر فضای مساله می‌شود، اما به هر حال در نهایت یک شبکه بیز با یک ساختار ثابت برای تمامی جمعیت ساخته می‌شود. حال اگر نوع و شکل روابط و وابستگی‌های موجود میان متغیرهای مساله مربوط به داده‌های منتخب در نواحی مختلف از فضای پاسخ با یکدیگر متفاوت باشد، آنگاه شبکه ساخته شده دارای کیفیت لازم و کافی نخواهد بود.

در نتیجه استفاده از یک ساختار در چنین مواقعی نه تنها ممکن است باعث از بین رفتن روابط منطقی میان ژن‌ها در یک کروموزوم گردد، بلکه می‌تواند منجر به همگرایی زودرس الگوریتم و افزایش احتمال افتادن در بهینه‌های محلی شود.

#### ۴- الگوریتم‌های خوشه‌بندی وقتی

با توجه به اینکه در این مقاله برای رفع مشکل الگوریتم بهینه‌سازی بیز استاندارد از الگوریتم‌های خوشه‌بندی وقتی به منظور افزایش فضای جستجو استفاده شده است، در این بخش مروری بر این دسته از الگوریتم‌های خوشه‌بندی خواهیم داشت.

مربوط به آن بلوک سازنده و با استفاده از راه‌حل‌های منتخب می‌سازد. سپس داده‌های مربوط به هر بلوک سازنده برای نسل بعد، با تولید اعداد تصادفی از مدل احتمالی شرطی مربوط به آن بلوک سازنده تولید می‌شود. این بلوک‌های سازنده در کنار یکدیگر قرار گرفته و نسل بعد را تولید می‌کنند. این مراحل تا آنجا که شرط توقف الگوریتم ارضا شود، تکرار می‌پذیرد. شرط توقف می‌تواند رسیدن به حداکثر تعداد اجرا و یا رسیدن به یک جواب قابل قبول باشد.



شکل ۳- یک شبکه بیز نمونه

آنچه الگوریتم بهینه‌سازی بیز را قدرتمند می‌سازد، در واقع وجود معیارهایی است که می‌توان صحت یک شبکه بیز را بر اساس آن سنجید. در نتیجه الگوریتم‌های حریصانه‌ای برای ساخت یک شبکه بیز به وجود آمده‌اند که می‌توانند از این معیارها برای ساخت و ارزیابی شبکه بیز استفاده نمایند. همچنین از دیگر نکات مثبت این الگوریتم این است که می‌تواند برای بهینه‌سازی مسایل در هر دو حوزه گسسته و پیوسته مورد استفاده قرار گیرد.

#### ۳- الگوریتم‌های تخمین توزیع برای محیط‌های

##### پیوسته

بهینه‌سازی در محیط‌های پیوسته با توجه به پیوسته بودن بازه مربوط به متغیرهای مساله مشکلات مربوط به خود را دارد. از این رو الگوریتم‌های تخمین توزیع متعددی برای بهینه‌سازی مسایل در محیط‌های پیوسته مطرح شده است. در این بخش شرح مختصری از الگوریتم‌های تخمین توزیع برای بهینه‌سازی توابعی در محیط‌های پیوسته ارائه شده است.

الگوریتم EGNA<sup>۲۱</sup> الگوریتمی است که مجموعه‌ای از راه‌حل‌های خوب را انتخاب کرده و سپس یک توزیع گوسی را بر اساس داده‌های انتخاب شده می‌سازد. نسل بعد نیز با تولید اعداد تصادفی از مدل گوسی ساخته شده تولید می‌شود [۲۵]. با توجه به اینکه EGNA الگوریتمی است که برای تمامی داده‌های منتخب یک توزیع گوسی تک‌قله‌ای می‌سازد، در نتیجه این الگوریتم برای مسایل بهینه‌سازی چند قله‌ای مناسب نمی‌باشد.

الگوریتم MBOA<sup>۲۲</sup>، تلاش در ساخت شبکه بیز برای راه‌حل‌های منتخب دارد که این شبکه بیز توسط یک درخت تصمیم نمایش داده می‌شود و هر درخت تصمیم نیز شامل اجزایی از نوع یک توزیع گوسی است. در نهایت تلاش می‌شود روابط میان متغیرهای یک مساله توسط درخت تصمیم نمایش داده شده مدل شود و سپس نسل بعد از شبکه ساخته شده نمونه‌برداری می‌شود. مشکل عمده این الگوریتم در واقع این است که درخت تصمیم با یک عمق معین فقط می‌تواند حد معینی از روابط بین متغیرها را نمایش دهد [۲۶].

الگوریتم mIDEA<sup>۲۳</sup>، نیز الگوریتمی است که تلاش می‌کند با استفاده از معیار اطلاعاتی بیز<sup>۲۴</sup> (BIC) یک مدل ترکیب گوسی<sup>۲۵</sup> را برای مدل کردن احتمالات راه‌حل‌های منتخب ساخته و سپس نسل بعد را از مدل احتمالی ساخته شده نمونه‌برداری نماید. مشکل الگوریتم mIDEA نیز در واقع این است که اگرچه با

## ۴-۱- الگوریتم خوشه‌بندی k'-means

$$dm(x_t, c_i) = \|x_t - c_i\|^2 - E \log_2(p_{C_i}) \quad (۶)$$

معیار (۶) ترکیبی از فاصله میان هر داده و مرکز خوشه مورد بررسی به همراه چگالی اطلاعاتی آن مرکز خوشه است که در آن  $x_t$  داده مورد نظر و  $c_i$  مرکز خوشه مورد بررسی و  $p_{C_i}$  نسبت تعداد داده‌های مرتبط با خوشه  $c_i$  به تعداد کل داده‌ها و  $E$  ثابتی برای کنترل میزان اثر بهره اطلاعاتی<sup>۲۶</sup> یک مرکز خوشه است. در پایان پس از تخصیص کلیه داده‌ها به خوشه‌های مربوطه، مرکز هر خوشه مجدداً تعیین می‌شود. معیار ارزیابی فوق سبب می‌گردد که خوشه‌هایی که دارای تعداد کمی داده متصل هستند به تدریج حذف گردند و با خوشه‌های دیگر ادغام شوند. این مراحل تا آنجا تکرار می‌گردد که خوشه نسبت داده شده به داده‌ها در دو توالی پی‌درپی تغییر نکند [۱۷، ۱۸].

الگوریتم خوشه‌بندی k-means یکی از الگوریتم‌های خوشه‌بندی است که به جهت کارایی و سادگی در استفاده کاربردهای زیادی دارد [۱۷]. این الگوریتم دارای یک روال تکراری است که برای تعدادی ثابت از خوشه‌ها تلاش در تخمین موارد زیر دارد:

۱. بدست آوردن نقاطی به عنوان مراکز خوشه‌ها: این نقاط در واقع همان میانگین نقاط متعلق به هر خوشه هستند.
  ۲. نسبت دادن هر نمونه داده به یک خوشه که آن داده کمترین فاصله تا مرکز آن خوشه را دارا باشد.
- مراحل کلی این الگوریتم در شکل ۵ نمایش داده شده است:

۱. انتخاب k نقطه به عنوان مراکز خوشه‌ها.
۲. انتساب هر نمونه داده به خوشه‌ای که مرکز آن خوشه کمترین فاصله تا آن داده را داراست.
۳. محاسبه مجدد مرکز هر خوشه پس از انتساب تمام داده‌ها به یکی از خوشه‌ها. (میانگین نقاط متعلق به هر خوشه)
۴. تکرار مراحل ۲ و ۳ تا زمانی که دیگر هیچ تغییری در مراکز خوشه‌ها حاصل نشود.

شکل ۵- شبه کد الگوریتم k-means

علی‌رغم اینکه خاتمه‌پذیری الگوریتم بالا تضمین شده است، ولی جواب نهایی آن یکتا نمی‌باشد. به طور کلی روش این روش دارای مشکلات زیر است:

- جواب نهایی به انتخاب خوشه‌های اولیه وابستگی دارد.
  - روالی مشخص برای محاسبه‌ی اولیه مراکز خوشه‌ها وجود ندارد.
  - اگر در تکراری از الگوریتم تعداد داده‌های متعلق به خوشه‌ای صفر شد، راهی برای تغییر و بهبود ادامه‌ی روش وجود ندارد.
  - در این روش فرض شده است که تعداد خوشه‌ها از ابتدا مشخص است. اما معمولاً در کاربردهای زیادی تعداد خوشه‌ها مشخص نمی‌باشد.
- بزرگترین مشکل الگوریتم k-means برای خوشه‌سازی بی‌نیاز به دانستن تعداد دقیق خوشه‌ها قبل از عمل خوشه‌بندی است. به عبارت دیگر در k-means، تعداد خوشه‌ها از قبل باید معلوم باشد؛ در [۱۷] یک الگوریتم وفقی مبتنی بر k-means که نیازی به تعیین تعداد دقیق خوشه‌ها از قبل ندارد، با نام k'-means، ارائه شده است. الگوریتم پیشنهادی متشکل از دو فاز اصلی است. در فاز اول حداکثر تعداد خوشه‌ها تعیین شده و با استفاده از الگوریتم k-means مراکز ابتدایی خوشه‌ها تعیین می‌شود و سپس این مراکز برای فاز دوم بکار می‌روند. در فاز دوم هر نمونه داده در فضا بر اساس یک معیار که ترکیبی از فاصله داده و مرکز هر خوشه است و همچنین چگالی داده مربوط به هر خوشه، به یک خوشه متصل می‌گردد. به طور کلی گام‌های این الگوریتم عبارت است از [۱۷، ۱۸]:

۱. ابتدا یک عدد  $k$  به عنوان حداکثر تعداد خوشه‌ها تعیین می‌شود. سپس کلیه داده‌ها با استفاده از الگوریتم k-means به  $k$  خوشه افراز می‌شوند. یعنی مرکز هر خوشه تعیین و سپس این مراکز به مرحله بعد وارد می‌شوند.

۲. در مرحله دوم برای هر خوشه یک چگالی داده ( $p_{C_i}$ ) که برابر با نسبت تعداد داده‌های مرتبط با آن خوشه به تعداد کل داده‌ها است، تعیین می‌شود. سپس مجدداً برای هر داده و هر مرکز خوشه، معیار (۶) محاسبه شده و هر داده به مرکز خوشه‌ای متصل می‌شود که کمترین مقدار را بر اساس معیار (۶) دارد.

## ۴-۲- الگوریتم خوشه‌بندی فازی وقفی

الگوریتم خوشه‌بندی فازی C-means یکی از الگوریتم‌های خوشه‌بندی است که کارایی خوبی را در حوزه‌های مختلف ارائه داده است. در این الگوریتم نمونه‌ها به  $C$  خوشه تقسیم می‌شوند که تعداد  $C$  از قبل مشخص شده است. در الگوریتم خوشه‌بندی فازی C-means تابع هدف بصورت زیر می‌باشد:

$$J = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d_{ik}^2 = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\| \quad (۷)$$

که  $m$  یک عدد حقیقی بزرگتر از ۱ است که در بیشتر موارد برای  $m$  عدد ۲ انتخاب می‌شود [۱۹].

اگر  $m$  برابر ۱ باشد، تابع هدف خوشه‌بندی C-means غیر فازی بدست می‌آید. همچنین  $x_k$  نمونه  $k$  ام و  $v_i$  نماینده یا مرکز خوشه  $i$  ام و  $n$  تعداد نمونه‌ها می‌باشد.  $u_{ik}$  میزان تعلق نمونه  $i$  ام در خوشه  $k$  ام را نشان می‌دهد. از روی  $u_{ik}$  می‌توان یک ماتریس  $U$  تعریف کرد که دارای  $C$  سطر و  $n$  ستون باشد و مولفه‌های آن مقداری بین ۰ تا ۱ را می‌توانند اختیار کنند. اگر تمامی مولفه‌های ماتریس  $U$  بصورت ۰ و یا ۱ باشند، الگوریتم مشابه C-means غیرفازی خواهد بود. با اینکه مولفه‌های ماتریس  $U$  می‌توانند مقداری بین ۰ تا ۱ را اختیار کنند، اما مجموع مولفه‌های هر یک از ستونها باید برابر ۱ باشد و داریم:

$$J = \sum_{i=1}^c u_{ik} = 1 \quad \forall \quad k = 1, 2, \dots, n \quad (۸)$$

مراحل کلی این الگوریتم در شکل ۶ نمایش داده شده است:

۱. مقدار دهی اولیه برای  $m$ ،  $C$  و  $U^0$  و تعیین خوشه‌های اولیه.
۲. محاسبه مراکز خوشه‌ها.
۳. محاسبه ماتریس تعلق از روی خوشه‌ها ( $U^1$ ).
۴. اگر تغییر در ماتریس  $U^1$  از یک حد تعیین شده کمتر بود، الگوریتم خاتمه می‌یابد و در غیر این صورت رفتن به مرحله ۲.

شکل ۶- شبه کد خوشه‌بندی فازی C-means

اما یکی از مشکلات این الگوریتم این است که تعداد خوشه‌ها از قبل باید معلوم باشد؛ در [۱۹] یک الگوریتم وفقی C-means که نیازی به تعیین تعداد دقیق خوشه‌ها از قبل ندارد، با نام FACT<sup>۲۷</sup>، پیشنهاد داده شده است. همانگونه

ندارد. بنابراین خوشه اضافه شده حذف و داده بعدی موجود در بردار Candidate برای خوشه جدید انتخاب می‌گردد [۱۹].

۳. در پایان اگر بهبودی با در نظر گرفتن تمامی داده‌های موجود در بردار Candidate صورت نگرفت، الگوریتم متوقف می‌شود. در غیر این صورت گام ۲ الگوریتم دوباره تکرار می‌شود [۱۹].

## ۵- الگوریتم پیشنهادی بهینه‌سازی بیز

در این بخش، یک الگوریتم بهینه‌سازی بیز ارائه می‌شود که در آن تلاش می‌شود برخی از مشکلات موجود در سایر الگوریتم‌های تخمین توزیع در محیط‌های پیوسته را برطرف سازد. از این رو در الگوریتم پیشنهادی تلاش می‌شود برای جستجوی بهینه فضا از چند شبکه بیز با ساختارهای متفاوت که در اثر افراز فضای جستجو به چند زیرفضا به وجود آمده‌اند، برای نمایش وابستگی‌های موجود استفاده شود. برای افراز فضای جستجو به چند زیرفضا، راه‌حل‌های مختلفی وجود دارد. یکی از راه‌های معمول و کارا استفاده از روش خوشه‌بندی است.

این روش باعث تقسیم‌بندی فضای مساله به تعدادی زیرفضا خواهد شد که این زیرفضاها بر اساس معیار شباهت تشکیل می‌شوند. بنابراین در الگوریتم پیشنهادی راه‌حل‌های انتخابی ابتدا با استفاده از یک روش خوشه‌بندی به چند خوشه افراز شده و سپس برای هر خوشه یک شبکه بیز تولید می‌گردد. بنابراین اگر شکل و نوع روابط و وابستگی‌های منطقی میان متغیرها در نواحی مختلف فضای جستجو با هم متفاوت باشد، در این صورت امکان وجود چندین شبکه بیز با ساختارها و پارامترهای مختلف وجود خواهد داشت که الگوریتم پیشنهادی قابلیت یافتن این شبکه‌ها را خواهد داشت. اما مساله دیگری که در این الگوریتم مطرح است تعیین روش مربوط به خوشه‌بندی است. لذا با توجه به این که در بیشتر الگوریتم‌های خوشه‌بندی نیاز به تعیین تعداد دقیق خوشه‌ها به عنوان یک پارامتر ورودی می‌باشد و همچنین با توجه به اینکه در بهینه‌سازی یک مساله ممکن است هیچ شهودی از فضای مورد بررسی از قبل وجود نداشته باشد و تعداد خوشه‌ها در مراحل مختلف جستجو متفاوت باشد، در نتیجه تعیین تعداد دقیق خوشه‌ها قبل از خوشه‌بندی کاری سخت و دشوار است. لذا برای حل این مشکل در این مقاله تلاش شده است برای خوشه‌بندی از الگوریتم‌های خوشه‌بندی وفقی که نیازی به تعیین تعداد دقیق خوشه‌ها به عنوان پارامتر ورودی ندارند و می‌توانند تعداد مناسب خوشه‌ها را در حین خوشه‌بندی به صورت خودکار تعیین کنند، استفاده شود.

در الگوریتم پیشنهادی در هر بار تکرار ابتدا  $T\%$  از بهترین راه‌حل‌ها بر اساس تابع برازش مساله به صورت قطعی انتخاب شده و سپس با استفاده از یک الگوریتم خوشه‌بندی وفقی، خوشه‌بندی شده و سپس برای هر خوشه یک شبکه بیز برای تخمین توزیع احتمال حاکم بر آن خوشه تولید می‌شود. ساختار و پارامترهای مربوط به شبکه بیز هر خوشه به صورت مستقل و بر حسب داده‌های مربوط به آن خوشه تعیین می‌شود. سپس به منظور جستجوی بهتر فضا و نیاز به تعداد کمتری از افراد به عنوان جمعیت اولیه،  $T\%$  از افراد نسل بعد توسط شبکه‌های بیز تولید شده و  $(100-T)\%$  دیگر به صورت تصادفی تولید می‌شوند. همچنین لازم به ذکر است، تعداد راه‌حل‌هایی که از هر شبکه بیز در نسل بعد قرار خواهد گرفت متناسب است با تعداد راه‌حل‌های قرار گرفته در خوشه مرتبط با آن شبکه بیز. این کار سبب می‌شود تا از شبکه‌های بیزی که افراد منتخب بیشتری دارند تعداد فرزندان بیشتری تولید گردد. این مراحل تا آنجا که شرط توقف الگوریتم ارضا شود، تکرار می‌پذیرد. در ادامه هر یک از مراحل الگوریتم شرح داده می‌شود. شبه کد این الگوریتم نیز در شکل ۷ نمایش داده شده است. در ادامه هر یک از بخش‌های الگوریتم پیشنهادی شرح داده می‌شود.

که واضح است در الگوریتم‌های خوشه‌بندی فازی هر نمونه داده تنها به یک خوشه متصل نمی‌شود؛ بلکه میزان انتساب هر نمونه داده به هر خوشه عددی بین ۰ تا ۱ است که هرچه این عدد برای یک نمونه داده و یک خوشه بیشتر باشد، در آن صورت احتمال انتساب داده مورد نظر به آن خوشه بیشتر است. اما الگوریتم FACT در واقع از چگالی داده مربوط به هر خوشه و تعیین نقاط پرت<sup>۲۸</sup> هر خوشه برای تعیین تعداد صحیح خوشه‌ها استفاده می‌کند. این الگوریتم دارای ۳ فاز اصلی می‌باشد. شرح این فازها در ادامه آمده است.

۱. الگوریتم فازی C-means را بر روی داده‌ها اجرا می‌کنیم. در نهایت دو داده زیر را در انتهای الگوریتم C-means خواهیم داشت:

- اگر تعداد خوشه‌ها C و تعداد داده‌ها N باشد آنگاه ماتریس  $C \times N$  بعدی U را که نشان دهنده میزان تعلق هر نمونه داده بر هر خوشه است، می‌سازیم.
- تعیین مقدار معیار چگالی خوشه<sup>۲۹</sup> (CDC) که برابر است با:

$$CDC = \sum_{i=1}^c \sum_{j=1}^n U_{i,j} \quad (9)$$

- ۲. در این فاز نقاط پرت محاسبه می‌شود. بدین منظور
- هر نقطه بر اساس ماتریس U به خوشه‌ای متصل می‌گردد که بیشترین احتمال را برای انتساب دارد. نتایج این انتساب بر اساس (۱۰) در ماتریس M ذخیره می‌شود.

$$M_{i,j} = \begin{cases} U_{i,j} & \text{if } \text{Max}_{i=1}^c (U_{i,j}) = U_{i,j} \\ 0 & \text{else} \end{cases} \quad (10)$$

- سپس این الگوریتم تلاش می‌کند نقطه پرت مربوط به هر خوشه را بر اساس (۱۱) تعیین می‌کند. بدین منظور نقطه پرت برای خوشه i برابر است با

$$\text{Candidate}_i = \text{Min}_{j=1}^n (M_{i,j}) \quad \text{if } M_{i,j} \neq 0 \quad (11)$$

- سپس در بین اعضای بردار Candidate کمترین مقدار انتخاب می‌شود. سپس اگر داده متناسب با این مقدار را  $OP = \{OP_1, OP_2, \dots, OP_r\}$  بنامیم، آنگاه یک خوشه جدید با مرکز (۱۲) اضافه می‌شود.

$$\text{Center}(c+1) = OP + \lambda \quad (12)$$

که  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_r\} \approx 0$  است.

- سپس مقدار U و CDC مجدداً بر اساس خوشه جدید و با اندیس t+1 محاسبه شده و سپس مقدار  $\theta_{t+1}$  بر حسب (۱۳) محاسبه می‌شود.

$$\theta_{t+1} = \alpha \left( \frac{CDC_{t+1}}{CDC_t} \right) + (1-\alpha) \theta_t \quad (13)$$

که در آن  $\alpha$  یک ضریب ثابت بین ۰ تا ۱ است.

حال چنانچه  $\theta_{t+1} > \theta_t$  خوشه جدید اضافه می‌شود، ولی اگر  $\theta_{t+1} \leq \theta_t$  آنگاه بدین معنی است که اضافه کردن خوشه جدید تأثیری در بهبود خوشه‌بندی

کل داده‌ها. به طوری که اگر تعداد داده‌های مربوط به خوشه  $j$  برابر با  $n_j$  باشد، آنگاه ضریب تاثیر  $B_j$  به صورت زیر تعریف می‌شود:

$$B_j = \frac{n_j}{\sum_{i=1}^{\psi} n_i} \quad (14)$$

این ضریب تاثیر به هنگام نمونه‌برداری از مدل و تولید افراد نسل بعد به کار گرفته می‌شود [۲۰].

## ۵-۲- ساخت شبکه‌های بیز

هدف از این مرحله ساخت یک شبکه بیز برای هر خوشه می‌باشد که بتواند با داده‌های موجود در آن خوشه همخوانی بهتری داشته باشد. یک شبکه بیز را با نماد  $M(\zeta, \theta)$  نمایش می‌دهیم که  $\zeta$  ساختار شبکه و  $\theta$  پارامترهای آن است. در حالت کلی برای ساخت یک شبکه بیز دو مساله مهم وجود دارد: معیار ارزیابی و رویه ساخت شبکه بیز [۲۱]:

- معیار ارزیابی شبکه: این معیار در واقع تعیین کننده میزان صحت شبکه ساخته شده با داده‌های مربوط به آن شبکه است.
- رویه ساخت: با توجه به اینکه ساخت شبکه بیز بهینه مرتبط با یک مجموعه داده، مساله‌ای تمام سخت [۲۸] است، در نتیجه انتخاب روش مناسب و کم‌هزینه برای ساخت شبکه بیز اهمیت می‌یابد.

## ۵-۲-۱- معیار ارزیابی شبکه بیز

برای ارزیابی شبکه بیز معیارهای متنوعی وجود دارد. در این مقاله برای ارزیابی شبکه بیز مربوط به یک خوشه از معیار اطلاعاتی بیز (BIC) که نمایانگر صحت شبکه برای تطابق با داده‌های مربوط به آن خوشه است به همراه معیاری برای ارزیابی پیچیدگی شبکه استفاده می‌گردد [۴، ۲۲، ۲۳]. معیار اطلاعاتی بیز به صورت زیر تعریف می‌شود:

$$BIC(M(\zeta_k, \theta_k)) = Accuracy(M(\zeta_k, \theta_k)) + Complexity(M(\zeta_k, \theta_k)) \quad (15)$$

که  $Accuracy(M(\zeta_k, \theta_k))$  نشان دهنده صحت شبکه بیز یک خوشه است که میزان تناسب ساختار شبکه و وابستگی‌های حاکم بر آن با داده‌های مربوط به آن خوشه را نمایش می‌دهد که برابر است با مجموع احتمالات شرطی راه‌حل‌هایی که در آن خوشه قرار گرفته‌اند و به صورت زیر تعریف می‌شود:

$$Accuracy(M(\zeta_k, \theta_k)) = \ln \left( \prod_{j=1}^{S_k} FM(\zeta_k, \theta_k)(Y^j) \right) \quad (16)$$

که  $|S_k|$  برابر است با تعداد راه‌حل‌های مرتبط با خوشه و  $FM(\zeta_k, \theta_k)(Y^j)$  برابر است با احتمال شرطی راه‌حل  $Y^j$ :

۱. تولید جمعیتی از راه‌حل‌های اولیه.
۲. محاسبه تابع برازش برای راه‌حل‌ها.
۳. انتخاب ۱٪ از بهترین راه‌حل‌ها به صورت قطعی بر اساس تابع برازش برای تولید شبکه بیز.
۴. خوشه‌بندی داده‌های منتخب با استفاده از الگوریتم‌های خوشه‌بندی افقی.
۵. ساخت یک شبکه بیز برای هر خوشه با توجه به داده‌های موجود در آن خوشه.
۶. تولید نسل بعد
a. تکرار به تعداد T% افراد جمعیت.
i. انتخاب یک خوشه با در نظر گرفتن ضریب تاثیر آن خوشه (B) به عنوان احتمال انتخاب شدن آن خوشه.
ii. تولید یک فرزند بر اساس شبکه بیز مرتبط با خوشه انتخابی.
b. تولید تصادفی $(1-T) \times 100$ افراد باقی‌مانده نسل بعد
۷. ادغام دو نسل بر اساس شایسته‌سالاری.
۸. خاتمه الگوریتم در صورت رسیدن به شرط خاتمه و در غیر این صورت رفتن به مرحله ۳.

شکل ۷- شبه کد الگوریتم پیشنهادی بهینه‌سازی بیز در محیط‌های پیوسته

## ۵-۱- خوشه‌بندی

هدف از این مرحله در واقع خوشه‌بندی راه‌حل‌های منتخب است. یکی از مشکلات الگوریتم استاندارد بهینه‌سازی بیز، استفاده و ساخت یک شبکه بیز برای مدل‌سازی تمامی راه‌حل‌های منتخب است؛ اگرچه استفاده از مدل ترکیب گوسی سبب جستجوی بهتر فضای مساله می‌شود، اما به هر حال در نهایت یک شبکه بیز برای تمامی داده‌ها ساخته می‌شود که این شبکه تنها یک ساختار معین و ثابت دارد. حال اگر روابط و وابستگی‌های موجود بر داده‌های منتخب در نواحی مختلف از فضای پاسخ با یکدیگر متفاوت باشد، آنگاه شبکه ساخته شده دارای کیفیت کافی نخواهد بود.

لذا در الگوریتم پیشنهادی ابتدا راه‌حل‌های انتخابی خوشه‌بندی شده و سپس برای هر خوشه یک شبکه بیز به صورت مستقل تولید می‌شود. این عمل باعث جستجوی بهتر فضا خواهد گردید. استفاده از خوشه‌بندی افقی سبب می‌شود که در ابتدای الگوریتم با توجه به بالا بودن تنوع جمعیت تعداد خوشه‌های تولید شده بیشتر باشد که این امر سبب جستجوی بهتر فضا می‌شود و در انتهای الگوریتم با توجه به پایین بودن تنوع جمعیت تعداد خوشه‌های تولید شده کمتر است و در نتیجه جستجو دقیق‌تر و متمرکزتر خواهد شد. در نهایت با توجه به این امر که همواره از خوشه بهتر تعداد بیشتری فرزند تولید شده و در جایگزینی، افراد بهتر همیشه باقی خواهند ماند و همچنین با توجه به ماهیت الگوریتم‌های تکاملی که تنوع جمعیت در طول تکامل کاهش می‌یابد، استفاده از خوشه‌بندی افقی به همگرایی الگوریتم کمک می‌کند.

یکی از مشکلات مرتبط با خوشه‌بندی این است که در بیشتر روش‌های خوشه‌بندی تعداد خوشه‌ها از قبل باید معلوم باشد؛ لذا در یک مساله بهینه‌سازی که هیچ شهودی از تعداد خوشه‌ها از قبل در دسترس نیست، استفاده از این روش‌ها ناکارآمد است. بدین منظور، در این مقاله از الگوریتم‌های خوشه‌بندی k'-means افقی و خوشه بندی فازی افقی که نیازی به تعیین تعداد دقیق خوشه‌ها قبل از خوشه‌بندی ندارند، استفاده شده است.

پس از اجرای الگوریتم خوشه‌بندی، راه‌حل‌های منتخب به  $\psi$  خوشه افزای می‌شوند. سپس به هر خوشه  $j$  یک ضریب تاثیر  $B_j$  نسبت داده می‌شود که این ضریب تاثیر متناسب است با نسبت تعداد داده‌های مرتبط با آن خوشه به تعداد

این مرحله تا آنجا ادامه می‌یابد که دیگر اضافه کردن هیچ یالی در معیار ارزیابی شبکه بهبودی ایجاد نکند [۲۱].

### ۵-۳- نمونه برداری از مدل

هدف از این مرحله تولید فرزندان برای نسل بعد می‌باشد. بدین منظور در این مقاله برای جستجوی بهتر فضای مساله در هر بار تکرار،  $T\%$  از فرزندان از مدل احتمالی ساخته شده و  $(100-T)\%$  دیگر به صورت تصادفی تولید می‌شوند. این امر سبب می‌گردد الگوریتم پیشنهادی برای شروع اجرا به جمعیت کمتری نیاز داشته باشد. همچنین با توجه به این که روش جایگزینی الگوریتم پیشنهادی بر مبنای شایسته‌سالاری است و افراد خوب همیشه در نسل بعد باقی می‌مانند، بنابراین تولید تصادفی افراد در هر نسل مانع از همگرایی الگوریتم پیشنهادی نخواهد شد. برای تولید هر فرد از نسل بعد ابتدا یک شبکه با احتمال  $B_j$  که برابر با ضریب تاثیر هر خوشه است، انتخاب شده و سپس درخت‌های موجود در شبکه بیز مورد نظر که معادل با بلوک‌های سازنده برای آن خوشه است، استخراج می‌شوند. سپس برای هر درخت موجود در شبکه انتخاب شده یک توزیع گوسی شرطی ساخته می‌شود که برگرفته از داده‌های مربوط به آن خوشه است و به صورت زیر تعریف می‌شود:

$$f(Y_i | \Pi_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \mu)^2}{2\sigma^2}}, \quad (21)$$

که  $f(Y_i | \Pi_i)$  احتمال تولید داده جزئی  $Y_i$  با وجود والد‌های  $\Pi_i$  است. همچنین

$$\sigma = \frac{1}{\sqrt{\sum_{i,i}^{-1}}}, \quad (22)$$

و  $\sum_{i,i}^{-1}$  معکوس ماتریس کواریانس میان داده‌های منتسب به خوشه مورد نظر است و

$$\mu = \mu_0 - \frac{\sum_{j=1}^n (\Pi_{j,i} - \mu_j) \sum_{j,i}^{-1}}{\sqrt{\sum_{i,i}^{-1}}}, \quad (23)$$

و  $n$  نیز تعداد عناصر مجموعه  $\Pi_i$  است. سپس برای تولید فرزند جدید، مقادیر هر بلوک سازنده با تولید عدد تصادفی از توزیع گوسی شرطی مرتبط با آن بلوک سازنده ساخته می‌شوند. بلوک‌های سازنده در کنار یکدیگر قرار گرفته و فرزند جدید را تشکیل می‌دهند. این مراحل تا آنجا ادامه پیدا می‌کند که  $T\%$  افراد نسل بعد تولید شوند. سپس، مابقی افراد به صورت تصادفی و با توزیع یکنواخت تولید می‌گردند.

### ۵-۴- جایگزینی

با توجه به تولید افراد تصادفی در هر نسل، الگوریتم پیشنهادی برای حفظ همگرایی از روش‌های شایسته‌سالاری در جایگزینی استفاده می‌کند تا افراد خوب همیشه زنده بمانند. بدین منظور، الگوریتم پیشنهادی، در هر مرحله پس از تولید افراد نسل بعد، آنها را با نسل فعلی ادغام نموده و کلیه افراد جمعیت را بر اساس

$$FM(\zeta_k, \theta_k)(Y) = \prod_{i=1}^d F_{\theta_k}(Y_i | \Pi_i), \quad (17)$$

که  $Y = (Y_1, Y_2, \dots, Y_d)$  یک بردار راه‌حل و  $\Pi_i$  والدین  $Y_i$  با توجه به ساختار  $\zeta_k$  و  $d$  نیز تعداد ابعاد مساله می‌باشد. واضح است که هرچه ساختار شبکه بیز تناسب بیشتری با داده‌های مربوطه داشته باشد، مقدار صحت برای شبکه بیز مورد ارزیابی بیشتر خواهد بود. پیچیدگی مدل،  $Complexity(M(\zeta_k, \theta_k))$ ، نیز عبارت است از:

$$Complexity(M(\zeta_k, \theta_k)) = \lambda \cdot \ln(|S_k|) | \mathcal{N}_k |, \quad (18)$$

که  $\lambda$  ضریبی برای تعیین میزان اثر پیچیدگی شبکه بر صحت و کارایی آن است. همچنین  $\mathcal{N}_k$  نیز پارامتری متناسب با پیچیدگی شبکه است که در این مقاله برای سهولت  $\mathcal{N}_k$ ، تعداد یال‌های حاکم بر شبکه در نظر گرفته شده است. در نهایت (۱۵) را می‌توان به صورت (۱۹) نوشت:

$$BIC(M(\zeta_k, \theta_k)) = -\ln\left(\prod_{j=1}^{|S_k|} F_{M(\zeta_k, \theta_k)}(Y^j)\right) + \lambda \cdot \ln(|S_k|) | \mathcal{N}_k |, \quad (19)$$

و از آنجا که  $-\ln\left(\prod_{j=1}^{|S_k|} F_{M(\zeta_k, \theta_k)}(Y^j)\right)$  برابر با نامحتمل‌ترین مقدار برای تابع احتمالاتی  $FM(\zeta_k, \theta_k)$  است، معیار ارزیابی در نهایت به صورت (۲۰) خواهد بود:

$$BIC(M(\zeta_k, \theta_k)) = |S_k| H(f_{M(\zeta_k, \theta_k)}(Y)) + \lambda \cdot \ln(|S_k|) | \mathcal{N}_k |, \quad (20)$$

که در آن  $H(f_{M(\zeta_k, \theta_k)}(Y))$  برابر با نامحتمل‌ترین مقدار برای  $f_{M(\zeta_k, \theta_k)}(Y)$  است [۴، ۵]. در نهایت هدف، ساخت شبکه بیزی است که این معیار را کمینه سازد.

### ۵-۲-۲- رویه جستجو

با توجه به اینکه ساخت شبکه بیز بهینه برای یک مجموعه داده، مساله‌ای تمام سخت است و تاکنون راه‌حل چندجمله‌ای برای آن یافت نشده است، به صورت معمول از یک الگوریتم حریصانه برای ساخت آن استفاده می‌شود [۲۱].

الگوریتم حریصانه تلاش می‌کند اعمال ممکن بر روی یک شبکه بیز را مانند اضافه کردن یک یال جدید، حذف یک یال، جابجایی دو یال را تا آنجا ادامه دهد که دیگر بهبودی در شبکه موجود حاصل نشود. بدین منظور در این مقاله تنها عملیات ممکن، اضافه کردن یک یال جدید به شبکه بیز در نظر گرفته شده است. در نهایت رویه ساخت، ابتدا با یک شبکه بدون یال آغاز کرده و در هر مرحله تلاش می‌کند یالی را بیفزاید که بیشترین بهبود را در معیار ارزیابی (۲۰) ایجاد می‌کند.

میزان برازش مرتب کرده و سپس به تعداد افراد جمعیت، بهترین آنها را گزینش می‌نماید.

## ۶- نتایج تجربی

در این بخش نتایجی از ارزیابی الگوریتم پیشنهادی ارائه شده است. تمامی این آزمایش‌ها با استفاده از پیاده‌سازی‌های کامپیوتری و با استفاده از نرم‌افزار Matlab صورت گرفته است. پارامترهای مورد استفاده برای آزمایش‌ها با استفاده از شبیه‌سازی‌های انجام گرفته تعیین شده و در جدول ۱ آمده است. ارزیابی‌ها بر اساس مجموعه توابع بهینه‌سازی CEC2005 که هم‌اکنون مرجع کاملاً معتبری برای ارزیابی یک الگوریتم بهینه‌سازی در محیط‌های پیوسته است و دارای انواع مختلفی از توابع بهینه‌سازی با شرایط متفاوت است، انجام گرفته است. این مجموعه دارای ۲۵ نمونه تابع بهینه‌سازی F1 تا F25 است که توابع F1 تا F6 توابع تک‌قله‌ای و توابع F6 تا F25 توابعی چندقله‌ای هستند.

جدول ۲ شماره، نام و دامنه این توابع را نمایش می‌دهد. ابعاد تمامی توابع ۳۰ در نظر گرفته شده است. همچنین با توجه به پارامتر  $F\_Bias$  که یک مقدار ثابت برای انتقال مقدار تابع است و ۰ در نظر گرفته شده است، نقطه بهینه سراسری برای تمامی توابع برابر با ۰ است. برای دریافت اطلاعات بیشتر درباره نام، تعریف، نوع، تعداد ابعاد، تصویر، شرایط و بهینه سراسری تمامی توابع موجود در این مجموعه می‌توان به مرجع [۲۴] مراجعه نمود. برای ارزیابی الگوریتم ارائه شده چندین آزمایش ترتیب داده شده است که در ادامه شرح داده می‌شود:

جدول ۱- مقدار پارامترهای بکاررفته در آزمایش‌ها  
الف) پارامترهای عمومی الگوریتم بیز

ردیف	نام پارامتر	مقدار
۱	ثابت $\lambda$	۰.۵
۲	تعداد افراد جمعیت	۱۰۰۰
۳	تعداد تکرار الگوریتم	۱۰۰۰۰۰
۴	تعداد والدین انتخابی (r)	۲۰٪ جمعیت
۵	F_Bias (برای توابع موجود در CEC2005)	۰
۶	ثابت T	۸۵٪ جمعیت

ب) پارامتر الگوریتم بهینه‌سازی بیز استاندارد

ردیف	نام پارامتر	مقدار
۱	تعداد اجزای ترکیبی برای مدل ترکیب گوسی	تعداد ابعاد تابع (۳۰)

ج) پارامترهای الگوریتم خوشه‌بندی k'-means و فقی

ردیف	نام پارامتر	مقدار
۱	تعداد خوشه‌های اولیه (l)	تعداد ابعاد تابع (۳۰)
۲	ثابت E	۰.۳

د) پارامترهای الگوریتم خوشه‌بندی فازی و فقی

ردیف	نام پارامتر	مقدار
۱	تعداد خوشه‌های اولیه (l)	تعداد ابعاد تابع (۳۰)
۲	ثابت $\alpha$	۰.۲

جدول ۲- توابع مورد آزمایش

شماره تابع	نام تابع	دامنه متغیرها
F1	Shifted Sphere Function	[-100 100]
F2	Shifted Schwefel's Problem	[-100 100]
F3	Shifted Rotated High Conditioned Elliptic Function	[-100 100]
F4	Shifted Schwefel's Problem with Noise in Fitness	[-100 100]
F5	Schwefel's Problem with Global Optimum on Bounds	[-100 100]
F6	Shifted Rosenbrock's Function	[-100 100]
F7	Shifted Rotated Griewank's Function without Bounds	[0 600]
F8	Shifted Rotated Ackley's Function with Global Optimum on Bounds	[-32 32]
F9	Shifted Rastrigin's Function	[-5 5]
F10	Shifted Rotated Rastrigin's Function	[-5 5]
F11	Shifted Rotated Weierstrass Function	[-0.5 0.5]
F12	Schwefel's Problem 2.13	$[-\pi \pi]$
F13	Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)	[-5 5]
F14	Shifted Rotated Expanded Scaffer's F6	بدون محدودیت
F15	Hybrid Composition Function	[-5 5]
F16	Rotated Hybrid Composition Function	[-5 5]
F17	Rotated Hybrid Composition Function with Noise in Fitness	[-5 5]
F18	Rotated Hybrid Composition Function	[-5 5]
F19	Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	[-5 5]
F20	Rotated Hybrid Composition Function with the Global Optimum on the Bounds	[-5 5]
F21	Rotated Hybrid Composition Function	[-5 5]
F22	Rotated Hybrid Composition Function with High Condition Number Matrix	[-5 5]
F23	Non-Continuous Rotated Hybrid Composition Function	[-5 5]
F24	Rotated Hybrid Composition Function	[-5 5]
F25	Rotated Hybrid Composition Function without Bounds	[2 5]

می‌روند. جدول ۴ میانگین خطای پاسخ الگوریتم نسبت به حالت بهینه و انحراف معیار آن را در طول ۵۰ اجرای مستقل برای الگوریتم پیشنهادی با خوشه‌بندی‌های k'-means وفقی و خوشه‌بندی فازی وفقی و الگوریتم بهینه‌سازی بیز استاندارد به نمایش در می‌آورد. همانگونه که در این نتایج مشهود است استفاده از خوشه بندی مناسب، جستجوی کارای فضا و ارایه نتایج بهتر را در اکثر توابع چندقله‌ای مورد بررسی نسبت به الگوریتم بهینه‌سازی بیز استاندارد به دنبال دارد. توابع F6 تا F25 همگی چند قلله‌ای هستند. تابع F6، تابعی بین توابع تک‌قله‌ای و چندقله‌ای است. نتایج بدست آمده حاکی از آن است که الگوریتم پیشنهادی با خوشه‌بندی فازی وفقی بهتر از الگوریتم بهینه‌سازی بیز استاندارد است. تابع F7، تابعی است که سختی آن در ابعاد پایین‌تر بیشتر است. در مورد این تابع نیز نتایج حاکی از آن است که الگوریتم پیشنهادی با خوشه‌بندی فازی وفقی و خوشه‌بندی k'-means وفقی بهتر از الگوریتم بهینه‌سازی بیز استاندارد است.

جدول ۴- میانگین خطا و انحراف معیار اجرای الگوریتم پیشنهادی برای توابع F6 تا F25

تابع	Fact BOA		k'-means BOA		BOA	
	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$
F6	1.93e+01	4.95e+01	1.20e+03	6.19e+02	3.05e+02	2.58e+03
F7	1.27e-03	1.32e+02	1.23e-02	1.45e+02	4.42e+00	3.59e+05
F8	3.76e-02	2.07e+02	4.58e-02	1.09e+01	1.13e-01	2.03e+02
F9	1.21e-01	6.80e+01	6.24e+00	2.39e+01	3.31e+01	2.40e+05
F10	4.89e+00	9.05e+01	4.05e+01	6.02e+01	5.34e+01	1.86e+02
F11	1.52e+00	3.11e+01	4.44e+00	1.80e+01	3.63e+00	2.95e+01
F12	1.39e+03	4.39e+03	1.33e+04	1.31e+04	3.41e+03	1.63e+03
F13	5.38e-01	3.96e+00	1.08e+00	3.58e+00	3.86e+00	1.13e+01
F14	7.81e-02	1.25e+01	2.68e-01	1.31e+01	1.94e-01	1.33e+00
F15	1.50e+01	3.56e+02	9.45e+01	3.68e+02	1.96e+01	8.72e+02
F16	2.99e+01	3.26e+02	4.01e+01	7.40e+01	8.10e+01	7.15e+01
F17	2.75e+01	2.79e+02	3.90e+01	8.53e+01	1.58e+02	1.56e+02
F18	3.23e+00	8.77e+02	1.12e+00	9.04e+02	1.60e+00	8.30e+02
F19	7.40e-01	8.80e+02	1.10e+00	9.04e+02	1.44e+00	8.34e+02
F20	8.47e-01	8.79e+02	1.19e+00	9.04e+02	1.32e+00	8.36e+02
F21	0.00e+00	5.00e+02	0.00e+00	5.00e+02	5.43e-01	8.54e+02
F22	1.69e+00	9.08e+02	1.55e+01	8.79e+02	4.84e+02	1.53e+03
F23	2.44+01	5.59e+02	3.44e-04	5.34e+02	8.07e-01	8.66e+02
F24	0.00e+00	2.00e+02	0.00e+00	2.00e+02	3.89e-01	2.13e+02
F25	4.48e-02	2.11e+02	6.70e-01	2.13e+02	5.50e-01	2.14e+02

## ۶-۱- آزمایش ۱: ارزیابی کارایی الگوریتم پیشنهادی بر روی توابع تک‌قله‌ای

هدف از این آزمایش، بررسی کارایی الگوریتم پیشنهادی روی توابع تک‌قله‌ای می‌باشد. در این آزمایش شش تابع تک‌قله‌ای F1 تا F5 برای ارزیابی به کار رفته است. جدول ۳ میانگین خطای پاسخ الگوریتم نسبت به حالت بهینه و انحراف معیار آن را در طول ۵۰ اجرای مستقل برای الگوریتم پیشنهادی با خوشه‌بندی‌های k'-means وفقی و (k'-means BOA) و خوشه‌بندی فازی وفقی (Fact BOA) و الگوریتم بهینه‌سازی بیز استاندارد (BOA) به نمایش در می‌آورد.

از آنجا که این توابع تک‌قله‌ای می‌باشند، بنابراین با توجه به نتایج حاصل شده می‌توان نتیجه گرفت که الگوریتم پیشنهادی کارایی بهتری را در مقایسه با الگوریتم بهینه‌سازی بیز استاندارد در مورد توابع تک‌قله‌ای ارایه می‌کند. توابع F1 تا F5، به گونه‌ای طراحی شده‌اند که به ترتیب تابع F1 آسان‌تر از تابع F2 و تابع F2 آسان‌تر از تابع F3 و ... است. همانگونه که از نتایج بر می‌آید، برای تابع F1 و تابع F2 الگوریتم پیشنهادی توانایی یافتن بهینه سراسری را دارد. برای تابع F3 نیز نتیجه الگوریتم پیشنهادی با خوشه‌بندی فازی وفقی بهتر از الگوریتم پیشنهادی با خوشه‌بندی k'-means وفقی و الگوریتم پیشنهادی با خوشه‌بندی k'-means وفقی است. یکی از خصوصیات تابع F4 وجود نویز در آن است که این امر عمل جستجو را مشکل می‌سازد. اما با این حال می‌توان مشاهده نمود که الگوریتم پیشنهادی با خوشه‌بندی k'-means وفقی قابلیت یافتن بهینه سراسری را دارد. برای تابع F5 نیز الگوریتم پیشنهادی با خوشه‌بندی فازی وفقی و خوشه‌بندی k'-means وفقی نتایج بهتری را در مقایسه با الگوریتم بهینه‌سازی بیز استاندارد ارایه می‌نماید.

جدول ۳- میانگین خطا و انحراف معیار اجرای الگوریتم پیشنهادی برای توابع F1 تا F5

تابع	Fact BOA		k'-means BOA		BOA	
	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$
F1	1.89e-10	9.36e-09	4.63e-10	9.35e-09	1.20e+02	7.05e-05
F2	7.47e-07	8.71e-06	1.49e-06	6.94e-07	4.01e+02	9.81e+02
F3	5.81e+04	8.77e+05	4.20e+05	1.10e+06	2.93e+06	7.45e+06
F4	8.55e+00	3.96e+01	1.74e-06	8.13e-07	3.42e+03	9.81e+03
F5	7.83e+01	2.18e+03	1.37e+03	4.23e+03	2.09e+03	9.91e+03

شکل ۸ نیز مقایسه الگوریتم پیشنهادی با استفاده از خوشه‌بندی k'-means وفقی (k'-means BOA) و خوشه‌بندی فازی وفقی (Fact BOA) و الگوریتم بهینه‌سازی بیز استاندارد (BOA) را برای تابع F1 نمایش می‌دهد.

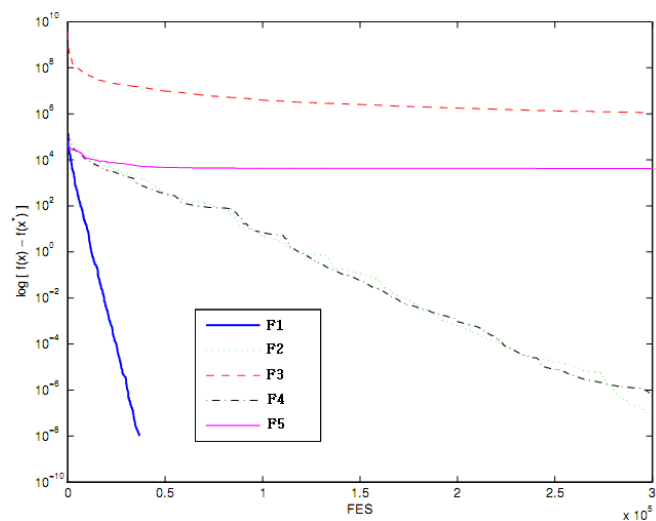
## ۶-۲- آزمایش ۲: ارزیابی کارایی الگوریتم پیشنهادی بر روی توابع چندقله‌ای

هدف از این آزمایش، بررسی کارایی الگوریتم پیشنهادی روی توابع چندقله‌ای می‌باشد. در این آزمایش نوزده توابع چندقله‌ای F6 تا F25 برای ارزیابی به کار

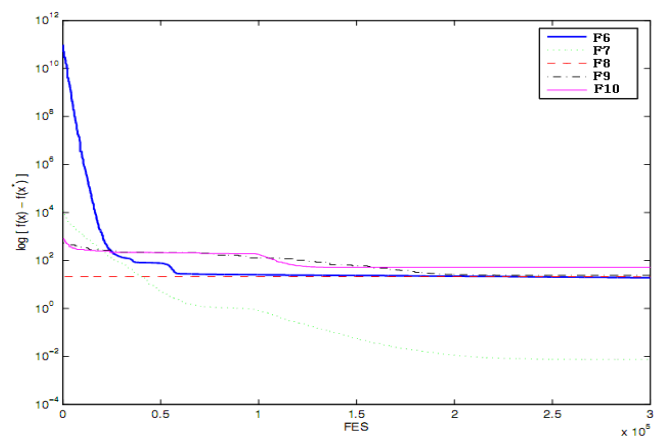
## ۴-۶- آزمایش ۴: بررسی تاثیر تغییر پارامترها در همگرایی الگوریتم پیشنهادی

در این آزمایش، اثر دو دسته پارامتر بر روی کارایی الگوریتم پیشنهادی بررسی می‌شوند. دسته اول به بررسی تاثیر تعداد متناسب خوشه‌ها (I) و تعداد راه‌حل‌های انتخابی (r) برای خوشه‌بندی در همگرایی الگوریتم می‌پردازد. بدین منظور تابع F11 را در نظر گرفته و مقدار r را که نمایش دهنده درصد راه‌حل‌هایی است که برای عمل خوشه‌بندی از میان بهترین راه‌حل‌های مساله بر اساس تابع براش انتخاب می‌شود، از ۱۰٪ تا ۶۰٪ تغییر داده شده است.

همچنین تعداد خوشه‌های اولیه مربوط به الگوریتم خوشه‌بندی k'-means وفقی و الگوریتم خوشه‌بندی فازی وفقی از ۲۰ تا ۱۰۰ تغییر داده شده است و سپس میزان تکرار لازم برای همگرایی الگوریتم را در اثر این تغییرات بررسی شده است. نمودار این بررسی برای الگوریتم خوشه‌بندی k'-means در شکل ۱۵ و برای الگوریتم خوشه‌بندی فازی در شکل ۱۶ نمایش داده شده است. همان‌گونه که در این اشکال مشهود است هرچه تعداد خوشه‌های اولیه (I) بیشتر باشد، الگوریتم وابستگی بیشتری به پارامتر r پیدا می‌کند ولی با کاهش تعداد خوشه‌ها مقدار پارامتر r آزادی بیشتری می‌یابد. همچنین در این اشکال، مناسب‌ترین مقدار برای پارامتر r برابر با ۲۰٪ و برای تعداد خوشه‌های اولیه حدودا بین ۲۰ تا ۳۰ می‌باشد.



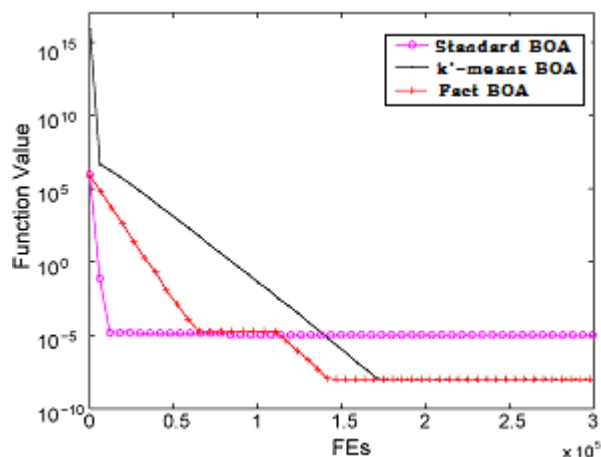
شکل ۱۰- نمودار همگرایی الگوریتم پیشنهادی برای توابع F1 تا F5



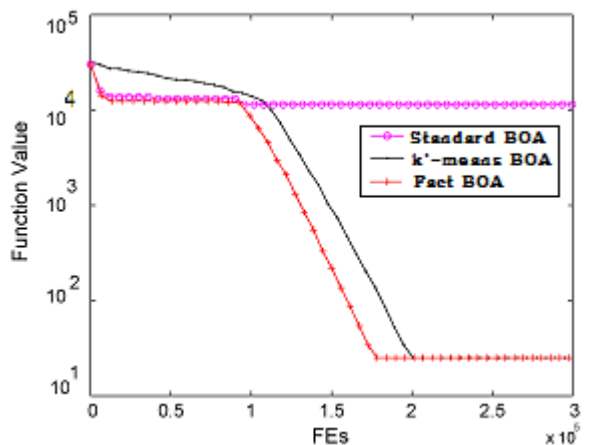
شکل ۱۱- نمودار همگرایی الگوریتم پیشنهادی برای توابع F6 تا F10

در مورد تابع F8، نیز الگوریتم پیشنهادی با خوشه‌بندی k'-means وفقی بهتر از الگوریتم بهینه‌سازی بیز استاندارد است. تابع F9 و تابع F10 نیز توابعی با بهینه‌های سراسری بسیار زیاد هستند. در این توابع نیز کارایی الگوریتم پیشنهادی با خوشه‌بندی فازی وفقی و خوشه‌بندی k'-means وفقی بهتر از الگوریتم بهینه‌سازی بیز استاندارد است. توابع F11 تا F14، نیز توابع توسعه داده شده از توابع پایه بهینه‌سازی و توابع F15 تا F25، توابعی قابل تجزیه به زیر فضاهای مستقل می‌باشند و همانگونه که قابل مشاهده است، نتایج بدست آمده در الگوریتم پیشنهادی در بیشتر موارد بهتر از الگوریتم بهینه‌سازی بیز استاندارد است.

شکل ۹ نیز مقایسه الگوریتم پیشنهادی با استفاده از خوشه‌بندی فازی وفقی (FACT BOA) و خوشه‌بندی k'-means وفقی (k-means BOA) را با الگوریتم بهینه‌سازی بیز استاندارد (BOA) برای تابع F9 نمایش می‌دهد.



شکل ۸- مقایسه الگوریتم پیشنهادی با الگوریتم بهینه‌سازی بیز استاندارد برای تابع F1

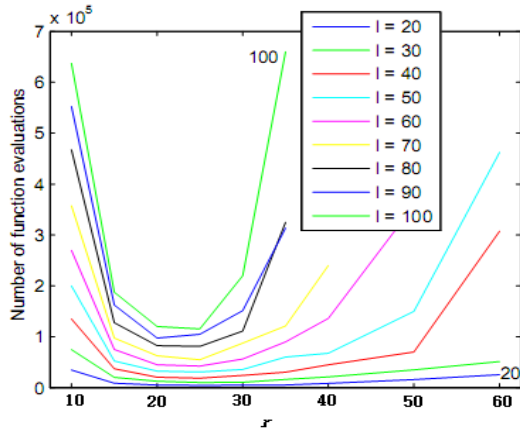


شکل ۹- مقایسه الگوریتم پیشنهادی با الگوریتم بهینه‌سازی بیز استاندارد برای تابع F9

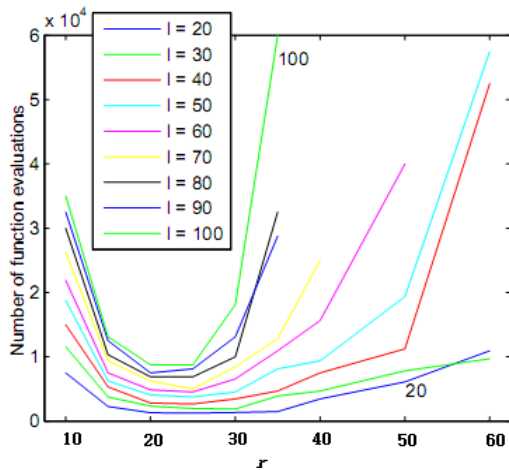
## ۳-۶- آزمایش ۳: ارزیابی همگرایی الگوریتم پیشنهادی

هدف از این آزمایش بررسی سرعت همگرایی الگوریتم پیشنهادی بر روی توابع F1 تا F25 است. بدین منظور همگرایی این توابع در شکل‌های ۱۰ تا ۱۴ نمایش داده شده است و همانگونه که در اشکال مشهود است، الگوریتم پیشنهادی پس از تعدادی تکرار به پایداری می‌رسد.

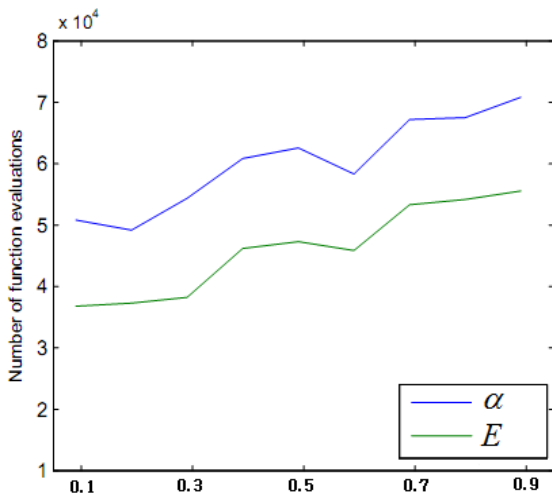
است با افزایش هر دو مقدار تعداد تکرار مورد نیاز برای همگرایی الگوریتم افزایش می‌یابد. همچنین شکل ۱۷ نشان می‌دهد، مناسب‌ترین مقدار برای پارامتر  $E$  برابر با ۰.۳ و برای  $\alpha$  حدوداً برابر با ۰.۲ می‌باشد.



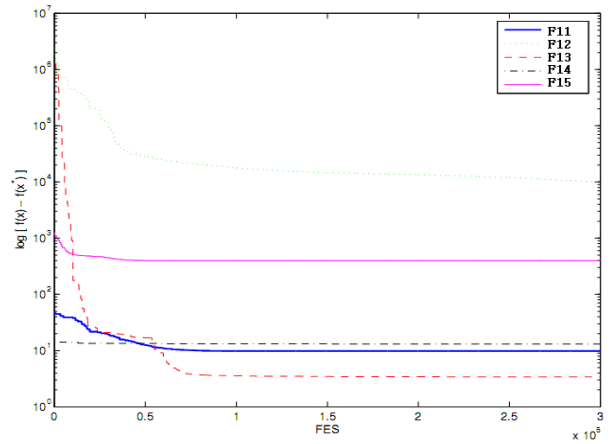
شکل ۱۵- نمودار تحلیل حساسیت برای پارامترهای درصد راه‌حل‌های انتخابی برای خوشه‌بندی ( $r$ ) و تعداد خوشه‌های اولیه در الگوریتم خوشه‌بندی  $k$ -means وقتی (۱)



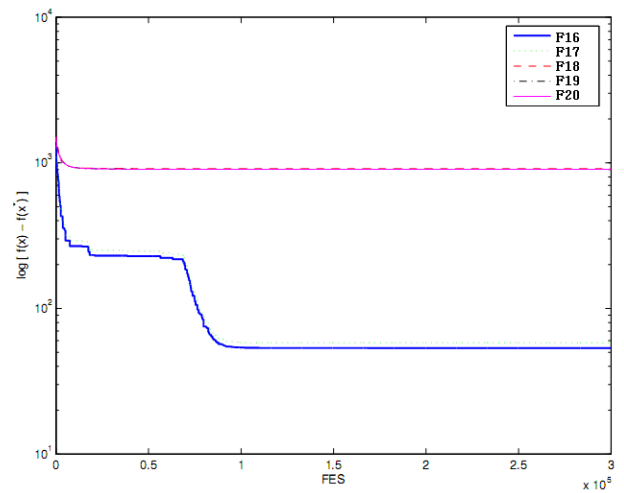
شکل ۱۶- نمودار تحلیل حساسیت برای پارامترهای درصد راه‌حل‌های انتخابی برای خوشه‌بندی ( $r$ ) و تعداد خوشه‌های اولیه در الگوریتم خوشه‌بندی فازی وقتی (۱)



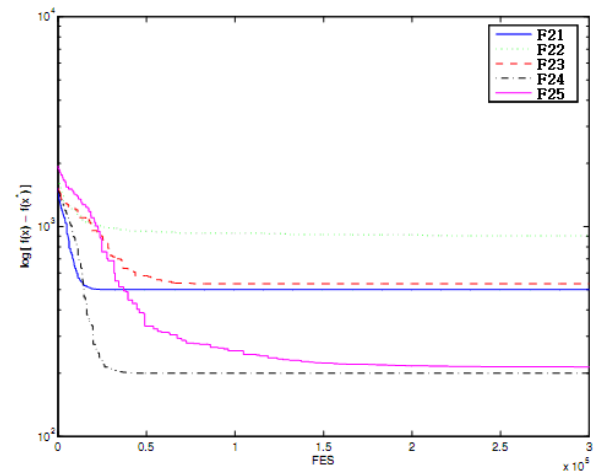
شکل ۱۷- نمودار تحلیل حساسیت برای پارامتر  $E$  در الگوریتم خوشه‌بندی  $k$ -means وقتی و پارامتر  $\alpha$  در الگوریتم خوشه‌بندی فازی وقتی



شکل ۱۲- نمودار همگرایی الگوریتم پیشنهادی برای توابع F11 تا F15



شکل ۱۳- نمودار همگرایی الگوریتم پیشنهادی برای توابع F16 تا F20



شکل ۱۴- نمودار همگرایی الگوریتم پیشنهادی برای توابع F21 تا F25

تحلیل دوم به منظور بررسی تاثیر تغییر در مقادیر اولیه ثابت  $E$  برای الگوریتم  $k$ -means وقتی و ثابت  $\alpha$  برای الگوریتم خوشه‌بندی فازی می‌پردازد. بدین منظور برای هر دو ثابت ذکر شده مقادیر را از ۰.۱ تا ۰.۹ تغییر داده و اثر این تغییر را در همگرایی الگوریتم پیشنهادی برای تابع F11 بررسی می‌کنیم. نمودار این بررسی در شکل ۱۷ نمایش داده شده است. همان‌گونه که در این شکل مشهود

## ۷- نتیجه گیری

[10] S. Baluja, and S. Davies, *Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space*, Proceedings of the International Conference on Machine Learning, pp. 30–38, 1997.

[11] M. Pelikan, and H. Muhlenbein, *The bivariate marginal distribution algorithm*, Advances in Soft Computing - Engineering Design and Manufacturing, pp. 521–535, 1999.

[12] H. Muhlenbein, and T. Mahnig, *FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions*, Evolutionary Computation, vol. 7, no. 4, pp. 353-376, 1999.

[13] G. Harik, *Linkage learning via probabilistic modeling in the ECGA*, IlliGAL Report No. 99010, Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 1999.

[14] M. Pelikan, D. E. Goldberg, and F. Lobo, *A survey of optimization by building and using probabilistic models*. Computational Optimization and Applications, vol. 21, no. 1, pp. 5-20, 2002.

[15] C. W. Ahn, R. S. Ramakrishna, and D. E. Goldberg, *Real-coded Bayesian optimization algorithm: Bringing the strength of boa into the continuous world*, GECCO (1), Lecture Notes in Computer Science, vol. 3102, pp. 840-851, Springer, 2004.

[16] C. W. Ahn, and R. Ramakrishna, *On the scalability of real-coded Bayesian optimization algorithms*, IEEE Transactions on Evolutionary Computation, vol. 12, no. 3, pp. 307–322, 2008.

[17] K. Zalik, *An efficient k-means clustering algorithm*. Pattern Recognition Letters vol. 29, pp. 1385–1391, 2008.

[18] B. Moradabadi, H. Beigy, and C. W. Ahn, *An improved real-coded Bayesian optimization algorithms*, IEEE Congress on Evolutionary Computation, 2011.

[19] F. Ensan, M. H. Yaghmaee, and E. Bagheri, *FACT: A new Fuzzy Adaptive Clustering Technique*, ISCC:IEEE Symposium on Computers and Communications, pp. 442-447, 2006.

[20] P. Lanzi, L. Nichetti, K. Sastry, D. Voltini, and D. Goldberg, *Real-Coded Extended Compact Genetic Algorithm Based on Mixtures of Models*. In Linkage in Evolutionary Computation, vol. 157, pp. 335-358, 2008.

[21] D. Heckerman, D. Geiger, and D. M. Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data*. Technical Report MSR-TR-94-09. Redmond, WA: Microsoft Research, 1994.

[22] J. A. Lozano, P. Larranaga, I. Inza, and E. (Eds.) Bengoetxea, *Towards a new evolutionary computation: Advances on estimation of distribution algorithms*. Berlin, Germany: Springer, 2006.

در این مقاله روشی برای بهبود الگوریتم بهینه‌سازی بیز در فضای پیوسته مطرح شده است که در آن تلاش شده است بر خلاف الگوریتم بهینه‌سازی بیز استاندارد که یک شبکه بیز سراسری را برای تخمین توزیع احتمال راه‌حل‌های منتخب می‌سازد و نسل بعد را از آن تولید می‌نماید، ابتدا راه‌حل‌های کاندید با استفاده از یک الگوریتم خوشه‌بندی وفقی که نیازی به دانستن تعداد دقیق خوشه‌ها از قبل ندارد، خوشه‌بندی شده و سپس یک شبکه بیز برای هر خوشه تولید و نسل بعد از این شبکه‌ها متولد گردد. این عمل جستجوی کارا تر را فضا برای توابعی که روابط داده‌ها در نقاط مختلف فضا با یکدیگر متفاوت است، نتیجه می‌دهد و نتایج شبیه‌سازی شده نیز نمایش دهنده برتری الگوریتم پیشنهادی با استفاده از الگوریتم خوشه‌بندی k'-means وفقی و الگوریتم خوشه‌بندی فازی وفقی بر الگوریتم بهینه‌سازی بیز استاندارد در محیط‌های پیوسته می‌باشد.

## مراجع

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley: Reading, MA, 1989.

[2] P. Larranaga, and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2002.

[3] M. Pelikan, D. E. Goldberg, E. and Cantu-Paz, *Linkage problem, distribution estimation, and Bayesian networks*, IlliGAL Report No. 98013. Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 1998.

[4] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, *BOA: The Bayesian optimization algorithm*, Proceedings of GECCO99, pp. 525-532, 1999.

[5] M. Pelikan, *Bayesian Optimization Algorithm: From Single Level to Hierarchy*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.

[6] H. Muhlenbein, and G. Paas, *From recombination of genes to the estimation of distributions*, Binary parameters. Parallel Problem Solving from Nature, pp. 178–187, 1996.

[7] S. Baluja, *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning*, Tech. Rep. No. CMU-CS-94-163. Pittsburgh, PA: Carnegie Mellon University, 1994.

[8] G. R. Harik, F. G. Lobo, and D. E. Goldberg, *The compact genetic algorithm*, IlliGAL Report No. 97006, Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 1997.

[9] J. S. De Bonet, C. L. Isbell, and P. Viola, *MIMIC: Finding optima by estimating probability densities*, Advances in neural information processing systems (NIPS-97), vol. 09, pp. 424–431, 1997.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۰/۲/۲۰

تاریخ اصلاح: ۹۰/۱۱/۳۰

تاریخ قبول شدن: ۹۰/۱۲/۲۵

نویسنده مرتبط: بهناز مرادآبادی، دانشکده مهندسی کامپیوتر، دانشگاه

صنعتی شریف، تهران، ایران.

- 1 Estimation of Distribution Algorithms
- 2 Evolutionary Algorithms
- 3 Bayesian Optimization Algorithms
- 4 Stochastic Optimization
- 5 Genetic Algorithms
- 6 Crossover
- 7 Mutation
- 8 Partial Solution
- 9 NP-Complete
- 10 Linkage Problem
- 11 Adaptive Clustering Algorithm
- 12 Univariate Marginal Distribution Algorithm
- 13 Population-Based Incremental Learning
- 14 Compact Genetic Algorithm
- 15 Greedy Algorithms
- 16 Mutual Information Maximization for Input Clustering
- 17 Combining Optimizers with Mutual Information Tress
- 18 Bivariate Marginal Distribution Algorithm
- 19 Factorized Distribution Algorithm
- 20 Extended Compact Genetic Algorithm
- 21 Estimaion of Gaussian Networks Algorithm
- 22 Mixed Bayesian Optimization Algorithm
- 23 Mixed Iterative-Density Estimation Algorithm
- 24 Bayesian Information Criterion
- 25 Gaussian Mixture Model
- 26 Information Gain
- 27 A New Fuzzy Adaptive Clustering Technique
- 28 Outlier
- 29 Cluster Density Criterion
- 30 Minimum likelihood

[23] C. W. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice*. Studies in Computational Intelligence. Springer-Verlag, Mar, 2006.

[24] *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real Parameter Optimization*, Accessed in <http://www.bionik.tu-berlin.de/user/niko/Tech-Report-May-30-05.pdf>.

[25] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, *Optimization in continuous domains by learning and simulation of Gaussian networks*, In A. S. Wu, editor, Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, pp. 201-204, 2000.

[26] J. Ocenasek, and J. Schwarz, *Estimation of distribution algorithm for mixed continuous discrete optimization problems*, In IOS Press, editor, 2nd Euro-International Symposium on Computational Intelligence, pp. 227-232, 2002.

[27] P. A. N. Bosman, Design and application of iterated density-estimation evolutionary algorithms, Doctoral Dissertation, Utrecht University, Utrecht, The Netherlands, 2003.

[28] D. M. Chickering, D. Geiger, and D. Heckerman, *Learning Bayesian network is NP-hard*, Technical Report MSR-TR-94-17, 1994.



**بهناز مرادآبادی** تحصیلات مربوط به مهندسی کامپیوتر را در دانشگاه تبریز در سال ۱۳۸۸ به اتمام رسانده است. سپس از سال ۱۳۸۸ تا سال ۱۳۹۰، تحصیل را در مقطع کارشناسی ارشد در رشته مهندسی کامپیوتر در دانشگاه صنعتی شریف ادامه داده و در آزمایشگاه هوش محاسباتی آن دانشگاه در مورد پردازش‌های تکاملی و خودکاره‌های یادگیر به تحقیق پرداخته است. زمینه‌های مورد علاقه ایشان پژوهش در مباحث مربوط به یادگیری ماشین، پردازش‌های تکاملی و یادگیری تقویتی می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:

behnaz.morady@gmail.com



**حمید بیگی** تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۷۱ و ۱۳۷۴ از دانشگاه شیراز و در مقطع دکتری مهندسی کامپیوتر در سال ۱۳۸۳ از دانشگاه صنعتی امیرکبیر به پایان رسانده است و هم‌اکنون دانشیار دانشکده مهندسی دانشگاه صنعتی شریف می‌باشد. زمینه‌های پژوهشی ایشان عبارتند از الگوریتم‌های موازی، الگوریتم‌های یادگیری، هوش محاسباتی و کاربردهای آن در شبکه‌های کامپیوتری می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

beigy@sharif.edu