استفاده از الگوریتم بهینه‌سازی گرگ خاکستری در خوشه‌یابی کلان‌داده‌ها

ایمان بهروان[۱]، دانشجوی دکتری؛ سید حمید ظهیری[۲]، استاد؛ سید محمد رضوی[۳]، دانشیار؛ روبرتو ترازارتی[۴]، محقق

۱- دانشکده مهندسی برق و کامپیوتر- دانشگاه بیرجند- بیرجند- ایران- i.behravan@birjand.ac.ir
۲- دانشکده مهندسی برق و کامپیوتر- دانشگاه بیرجند- بیرجند- ایران- hzahiri@birjand.ac.ir
۳- دانشکده مهندسی برق و کامپیوتر- دانشگاه بیرجند- بیرجند- ایران- smrazavi@birjand.ac.ir
۴- آزمایشگاه استخراج اطلاعات و داده‌کاوی- موسسه علوم و فناوری اطلاعات- پیزا- ایتالیا- roberto.trasarti@isti.cnr.it

چکیده: امروزه حجم بسیار زیادی از اطلاعات و داده‌ها از منابع مختلف نظیر گوشی‌های هوشمند، شبکه‌های اجتماعی، تکنولوژی‌های عکاسی و سایر منابع تولید می‌شود. بررسی و پردازش این حجم عظیم از اطلاعات چالش دهه‌های اخیر است که به آن کلان‌داده گفته می‌شود. یکی از روش‌های پرکاربرد استخراج اطلاعات، خوشه‌یابی است. خوشه‌یابیِ کلان‌داده‌ها چالش بزرگی است که توجه بسیاری از محققین را به خود جلب کرده است. در این پژوهش ابتدا یک روش خوشه‌یابی غیر خودکار (برای حالتی که تعداد خوشه‌ها از قبل مشخص است) و سپس یک روش خوشه‌یابی خودکار (قادر به یافتن تعداد خوشه‌ها) با استفاده از الگوریتم بهینه‌سازی گرگ خاکستری برای خوشه‌یابی کلان‌داده‌ها ارائه شده است. روش خوشه‌یابی خودکار یک روش دو مرحله‌ایست که در مرحله‌ی اول یک ساختار درخت گونه از الگوریتم مورد نظر برای یافتن تعداد خوشه‌ها اجرا می‌شود و در مرحله‌ی دوم الگوریتم اصلی فضا را برای یافتن موقعیت مراکز خوشه‌ها جست‌وجو می‌کند. عملکرد روش ارائه شده بر روی ۱۳ مجموعه داده‌ی مصنوعی و ۲ مجموعه کلان‌داده‌ی واقعی مربوط به مسیرهای طی شده توسط خودروها در سطح شهر پیزا مورد ارزیابی قرار گرفته و نتایج آن بررسی شده است. نتایج به دست آمده نشان از دقت بالای این الگوریتم در خوشه‌یابی داده‌های بزرگ و حجیم دارد.

واژه‌های کلیدی: کلان‌داده، خوشه‌یابی خودکار، روش‌های هوش جمعی، الگوریتم بهینه‌سازی گرگ خاکستری.

Using Grey Wolf Optimization Algorithm in Big Data Clustering

Iman Behravan[1], PhD student; Seyed Hamid Zahiri[2], Full Professor; Seyed Mohammad Razavi[3], Associate Professor; Roberto Trasarti[4], Researcher

1- Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran, Email: i.behravan@birjand.ac.ir
2- Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran, Email: hzahiri@birjand.ac.ir
3- Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran, Email: smrazavi@birjand.ac.ir
4- KDD lab, ISTI-CNR, Pisa, Italy, Email: roberto.trasarti@isti.cnr.it

Abstract: The huge amount of data created constantly with increasing rate from different sources such as smart phones, social media, imaging technologies and etc. becomes difficult to be analyzed by conventional data analytic tools. For this reason a new field of research called Big Data Analytics is growing faster in the research and industrial communities. Clustering big datasets is one of the important challenges which attracts more and more attentions among researchers. In this paper first a method for non-automatic big data clustering (when the number of clusters is known) and then a two-stage method for big data automatic clustering (able in finding the number of clusters) based on grey wolf optimization algorithm are introduced. In the first stage the algorithm tries to find the number of clusters using a tree structure and in the second stage the main algorithm searches the solution space to find the position of centroids. The methodology is tested on 13 synthetics and 2 real big mobility datasets. The achieved results show its effectiveness in big data clustering.

Keywords: Big data, Automatic clustering, Swarm intelligence methods, Grey wolf optimization algorithm.

تاریخ ارسال مقاله: ۱۳۹۷/۰۲/۲۳
تاریخ اصلاح مقاله: ۱۳۹۷/۰۵/۳۰و ۱۳۹۷/۰۸/۲۵
تاریخ پذیرش مقاله: ۱۳۹۷/۱۲/۰۱
نام نویسنده مسئول: سید حمید ظهیری
نشانی نویسنده مسئول: دانشکده مهندسی برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.

۱- مقدمه

امروزه حجم غیر قابل باوری از اطلاعات با نرخ فزاینده و از منابع مختلف همانند گوشی‌های هوشمند، شبکه‌های اجتماعی، تکنولوژی‌های عکاسی و منابع مختلف دیگر تولید می‌شود. پردازش و استخراج اطلاعات با معنی از این حجم عظیم از داده‌ها توسط روش‌های معمول تحلیل داده، امری دشوار به نظر می‌رسد. این امر باعث ظهور یک زمینه‌ی تحقیقاتی جدید به نام تحلیل کلان‌داده‌ها[۱] [۱] شده و توجه محققین زیادی را به خود جلب نموده است.

کلان‌داده به مجموعه داده‌هایی اطلاق می‌شود که دارای ویژگی‌های خاصی هستند از جمله: حجم زیاد داده‌ها، تعداد زیاد ویژگی‌ها و سرعت زیاد رشد داده‌ها. این ویژگی‌ها باعث پیدا شدن چالش‌های متعددی در تحلیل آن‌ها شده است که رفع این چالش‌ها مستلزم ارائه‌ی الگوریتم‌های جدید برای هر کدام از آن‌ها می‌باشد. به طور کلی منظور از کلان‌داده، مجموعه داده‌ایست که پردازش آن با استفاده از روش‌های سنتی امری بسیار دشوار است. تحلیل کلان‌داده‌ها، استخراج اطلاعات با معنی از کلان‌داده‌ها از طریق ایجاد مدل‌های یادگیری است[۲، ۳]. چالش‌های موجود در این حوزه نظیر حجم زیاد داده‌ها، تغییرات دینامیکی آن‌ها و وجود نویز باعث دشوار شدن کشف و استخراج الگوهای معنی‌دار مخفی در کلان‌داده‌ها شده است. از این رو الگوریتم‌های جدید و کارآمدی برای این حوزه باید ارائه شوند. خوشه‌یابی، به عنوان یک روش داده‌کاوی مهم، فرایند گروه‌بندی داده‌ها بر اساس شباهت‌های آن‌هاست[۶-۴]. این روش در زمینه‌های مختلفی نظیر زیست شناسی، علوم اجتماعی، بازاریابی و سایر زمینه‌ها حائز اهمیت است. در حقیقت هدف از خوشه‌یابی تقسیم نمونه‌های موجود در یک مجموعه داده به گروه‌های (یا خوشه‌های) مختلف است به نحوی که نمونه‌های موجود در یک گروه بیش‌ترین شباهت و نمونه‌های قرار گرفته در گروه‌های مختلف کم‌ترین شباهت را به هم داشته باشند. تاکنون تکنیک‌های مختلفی برای خوشه‌یابی معرفی شده‌اند از جمله *Kmeans* [۷] و *fuzzy-cmeans* [۸]. این روش‌های متداولِ خوشه‌یابی در مواجهه با کلان‌داده‌ها عملکرد مناسبی از لحاظ دقت اجرا نخواهند داشت که دلیل آن تعداد بسیار زیاد نمونه‌های موجود در این‌گونه مجموعه داده‌هاست [۹].

در این پژوهش از یک روش ابتکاری برای حل مساله‌ی خوشه‌یابی کلان‌داده‌ها استفاده شده است. این روش‌ها ابزار بسیار مفیدی در حل مسائل بسیار پیچیده‌ی مهندسی هستند. اینگونه روش‌ها که جایگزین روش‌های ریاضی مبتنی بر مشتق (نظیر لاگرانژین) شده‌اند، برای حل مسائل پیچیده با ابعاد بسیار زیاد که روش‌های ریاضی قادر به حل آن نمی‌باشند، ارائه شده‌اند. توانایی این روش‌ها در جست‌وجوی مؤثر فضای پاسخ نکته‌ایست که آن‌ها را به یک گزینه‌ی بی بدیل در حل مسائل پیچیده‌ی بهینه‌سازی تبدیل می‌کند. مسائلی که نه تنها دارای ابعاد بسیار زیادی هستند بلکه دارای نقاط بهینه‌ی محلیِ زیادی نیز می‌باشند و حل کردن آن‌ها نیازمند استفاده از روش‌هایی است که توانایی فرار کردن از این نقاط بهینه‌ی محلی را داشته باشند.

خوشه‌یابی خودکار کلان‌داده‌ها از جمله‌ی این مسائل پیچیده است که حل کردن آن با استفاده از روش‌های مرسوم این حوزه به دلیل تعداد زیاد داده‌ها و همچنین تعداد ویژگی‌های زیاد کاری دشوار است. روش‌های سنتی خوشه‌یابی دارای نقاط ضعفی هستند که کاربرد آن‌ها را برای کلان‌داده‌ها مشکل می‌کند از جمله: گیرافتادن در نقاط بهینه‌ی محلی، تولید پاسخ‌های اولیه‌ی تصادفی و عدم توانایی در پیدا کردن تعداد خوشه‌ها. از این رو برای خوشه‌یابی کلان‌داده‌ها باید از روش‌هایی استفاده کرد که نقاط ضعف ذکر شده را پوشش دهند.

روش‌های هوش جمعی، که جزو روش‌های ابتکاری بوده و الهام گرفته شده از فرایندهای موجود در طبیعت هستند ابزارهای مناسبی برای حل مساله‌ی خوشه‌یابی کلان‌داده‌ها می‌باشند [۱۰]. این روش‌ها از یک جمعیت برای جست‌وجوی فضای پاسخِ مساله‌ی بهینه‌سازی مورد نظر، جهت یافتن پاسخ بهینه، استفاده می‌کنند که هر عضو این جمعیت یک پاسخ احتمالی به مساله‌ی بهینه‌سازی مورد نظر می‌باشد. علاوه بر این، این روش‌ها دارای پارامترهایی برای کنترل فرایند جست‌وجو می باشند. در حقیقت روش‌های مورد نظر به کمک این پارامترها قادر هستند از نقاط بهینه‌ی محلی فرار کرده و فضای پاسخ را به صورت مؤثر جست‌وجو نمایند. این قابلیت، که باعث برتری آن‌ها بر روش‌های خوشه یابی سنتی می‌شود، دارای اهمیت زیادی در خوشه‌یابی کلان‌داده‌ها است. همچنین امکان جست‌وجوی فضای پاسخ برای یافتن تعداد خوشه‌ها، نقطه‌ی قوت دیگر این روش‌ها می‌باشد. به عبارت دیگر با تبدیل مساله‌ی خوشه‌یابی کلان‌داده‌ها به یک مساله‌ی بهینه‌سازی، می توان از روش‌های هوش جمعی برای حل این مساله استفاده نمود. در این پژوهش یک روش خوشه‌یابی خودکار با استفاده از الگوریتم ابتکاری گرگ خاکستری[۱۱] جهت خوشه‌یابی کلان‌داده‌ها ارائه شده است.

روش ارائه شده توانایی بالایی در جست‌وجوی فضای پاسخ برای یافتن نقطه‌ی بهینه‌ی کلی و یافتن تعداد خوشه‌ها دارد. جهت ارزیابی عملکرد روش پیشنهادی، دقت روش مورد نظر روی ۱۳ مجموعه‌داده‌ی مصنوعی با تعداد نمونه‌ها و تعداد ابعاد قابل قبول و همچنین ۲ مجموعه کلان‌داده‌ی واقعی تست شده و مورد بررسی قرار گرفته است. به طور کلی نوآوری این پژوهش عبارت است از: ۱- استفاده از الگوریتم ابتکاری گرگ خاکستری برای اولین بار در خوشه‌یابیِ کلان‌داده‌ها. ۲- ارائه‌ی یک روش خودکار در خوشه‌یابی کلان‌داده‌ها که با دقت بالایی قادر به یافتن تعداد خوشه‌های مجموعه داده‌ی مورد نظر می‌باشد. ساختار این مقاله بدین صورت است: بخش دوم مروری بر کارهای انجام شده در این زمینه می‌باشد. در بخش سوم الگوریتم بهینه‌سازی گرگ خاکستری معرفی خواهد شد. بخش‌های چهارم و پنجم به ترتیب به معرفی روش ارائه شده و ارائه‌ی نتایج اختصاص دارند. در بخش ششم نیز نتایج مربوط به داده‌های واقعی مورد بررسی قرار گرفته است. بخش آخر هم مربوط به نتیجه‌گیری می‌باشد.

۲- تاریخچه

۲-۱ - استفاده از روش‌های سنتی در خوشه‌یابی کلان‌داده‌ها

در[۱۲] یک روش خوشه‌یابی مبتنی بر الگوریتم سنتی *Kmeans* برای کلان‌داده‌ها ارائه شده است. این روش که به صورت موازی با استفاده از *Spark* پیاده سازی شده است، دارای دو فاز است. در فاز اول به تعداد k_{max} (حداکثر تعداد ممکن خوشه‌ها) با استفاده از یک تابع احتمال (به جای تصادفی انتخاب کردن خوشه‌ها) مرکز خوشه انتخاب می‌شود. این کار با هدف تولید پاسخ‌های اولیه‌ی بهتر و کاهش احتمال گیرافتادن در نقطه‌ی بهینه‌ی محلی انجام می‌شود. در فاز بعدی الگوریتم *Kmeans* به صورت موازی و بر روی ماشین‌های مختلف اجرا شده و k_{max} مرکز خوشه را پیدا می‌کند. پس از آن جهت یافتن تعداد دقیق خوشه‌ها، مراکز خوشه‌ی نزدیک به هم ادغام می‌شوند. بدین صورت که برای هر جفت مرکز خوشه، فاصله‌ی نمونه‌های متعلق به آن‌ها از هم محاسبه شده و در نهایت میانگین آن‌ها به عنوان یک مقدار آستانه در نظر گرفته می‌شود. پس از آن اگر فاصله‌ی دو مرکز خوشه از هم کمتر از این مقدار آستانه باشد خوشه‌های مورد نظر ادغام می‌شوند. بنابر ادعای نویسندگان، این روش تعداد خوشه‌ها را به درستی پیدا می‌کند اما روش آن‌ها دارای نقاط ضعف بزرگی است از جمله: ۱- باید k_{max}به عنوان یک پارامتر ورودی توسط کاربر وارد شود که تخمین زدن آن برای کلان‌داده‌ها دشوار است. اگرچه روابطی برای تخمین زدن k_{max} تاکنون معرفی شده است، اما قطعا بین k_{max} و تعداد نمونه‌های مجموعه داده رابطه‌ی مستقیمی وجود دارد. در نتیجه برای کلان‌داده‌ها k_{max} عدد بزرگی خواهد بود و در این روش باید تعداد زیادی مرکز خوشه تولید شده و در آخر ادغام شوند که کار پر هزینه‌ایست و می‌تواند منجر به کاهش دقت روش ارائه شده شود. ۲- روش ارائه شده بر روی مجموعه داده‌های معمولی با تعداد نمونه‌ها و تعداد ابعاد کم و همچنین خوشه‌های کاملاً قابل تفکیک و جدا از هم مورد ارزیابی قرار گرفته است و در مورد مجموعه داده‌هایی که در آن‌ها خوشه‌ها همپوشانی دارند ادعایی نشده است. پس نمی‌توان به ارزیابی درستی از روش ارائه شده رسید. ۳- ادغام کردن خوشه‌هایی که از هم فاصله دارند و کاملاً مجزا هستند با استراتژی معرفی شده در این پژوهش کار راحتی است اما در مورد خوشه‌هایی که مرز مشخصی ندارند و به راحتی قابل تفکیک نیستند، سخت و چالش برانگیز است.

در یک پژوهش مشابه، جین و همکارانش یک روش خوشه‌یابی غیرخودکار از طریق اصلاح الگوریتم Kmeans معرفی کردند [۱۳]. در این روش ابتدا از بین M بردار ویژگی موجود در مجموعه داده، m بردار ویژگیِ با اهمیت‌تر انتخاب شده و براساس اهمیتی که در مساله‌ی مورد بررسی دارند مرتب می‌شوند. یعنی در هر نمونه ویژگی اول با اهمیت‌ترین ویژگی در میان ویژگی‌های انتخاب شده است. سپس در مرحله‌ی بعد پراکندگی داده‌ها در هر بعد (یا ویژگی) محاسبه می‌شود. پس از آن نمونه‌ها بر اساس مقدار ویژگی اول به یک خوشه از میان k خوشه نسبت داده می‌شوند. این کار با استفاده از رابطه‌ی زیر انجام می‌شود:

$$\min_1 + j \times \delta_1 \le value_1 < \min_1 + (j+1) \times \delta_1 \qquad (۱)$$

در رابطه‌ی فوق δ_1 میزان پراکندگی داده‌ها در بعد اول، min_1 کم‌ترین مقدار داده‌ها در بعد اول، j شماره‌ی خوشه و $value_1$ مقدار ویژگی اول برای داده‌ی مورد نظر می‌باشد. اگر این مقدار در نامساوی فوق صدق کند، داده‌ی مورد نظر به خوشه‌ی jام نسبت داده می‌شود. پس از نسبت دادن نمونه‌ها به خوشه‌های مختلف مراکز خوشه‌ها با میانگین گرفتن مشخص می‌شوند. در مرحله‌ی بعد داده‌های پرت با استفاده از سایر ویژگی‌ها مشخص شده و مراکز خوشه‌ها به‌روزرسانی می‌شوند. مزیت اصلی این روش توانایی آن در پیدا کردن خوشه‌های با شکل پیچیده است. اما ایراد بزرگ آن، عدم توانایی در پیدا کردن تعداد خوشه‌ها می باشد. این روش نیز بر روی مجموعه داده‌هایی با خوشه‌های کاملاً جدا از هم مورد ارزیابی قرار گرفته است.

در[۱۴] یک روش خوشه‌یابی دیگر به نام *DisK-means* برای کلان‌داده‌ها معرفی شده است. در این روش هدف اصلی تولید پاسخ‌های اولیه‌ی با کیفیت جهت تولید پاسخ نهایی بهتر بوده است. در این روش که به صورت موازی بر روی بستر *Hadoop* نیز پیاده سازی شده است، ابتدا مجموعه داده‌ی مورد بررسی به m زیرمجموعه تقسیم می‌شود. سپس برای هر زیرمجموعه ماتریس *M(i,j)* که شامل فاصله‌ی هر نمونه از سایر نمونه‌های موجود در خوشه می‌باشد، محاسبه شده و نقطه‌ای که جمع فواصل آن کم‌ترین مقدار را داشته باشد به عنوان نقطه‌ی اولیه برای الگوریتم *Kmeans++* انتخاب می‌شود. پس از اجرای الگوریتم *Kmeans++* کیفیت m پاسخ ارائه شده توسط این الگوریتم با استفاده از تابع توان دوم مجموع فواصل اندازه‌گیری می‌شود. پاسخ با کیفیت‌تر به عنوان پاسخ اولیه برای مرحله‌ی آخر که الگوریتم *Kmeans* اجرا می شود، انتخاب می‌گردد. در این روش اگرچه زمان اجرا کاهش خوبی داشته است اما ایراد بزرگ روشِ *Kmeans* که عدم توانایی در پیدا کردن تعداد خوشه‌ها می‌باشد، کماکان وجود دارد. این ایراد قابل استفاده بودنِ روش ارائه شده برای کلان‌داده‌ها را زیر سؤال می‌برد. علاوه بر این، روش *Kmeans++* تنها تفاوتی که با روش سنتی *Kmeans* دارد در انتخاب مراکز اولیه است [۱۵]. این روش در ابتدا کاملاً به صورت تصادفی یک مرکز خوشه انتخاب می‌کند، پس از آن سایر مراکز به ترتیب و با استفاده از یک تابع احتمال انتخاب می‌شوند. یعنی احتمال انتخاب هر مرکز خوشه به مراکز انتخاب شده‌ی قبلی بستگی دارد. این یعنی برای انتخاب k مرکز خوشه‌ی اولیه تعداد k تکرار باید انجام شود که این یک نقطه ضعف برای روش ارائه شده به خصوص در مواجهه با کلان‌داده‌ها می باشد.

۲-۲- استفاده از روش‌های ابتکاری در خوشه‌یابی کلان‌داده‌ها

در سال ۲۰۱۵ روشی تحت عنوان *Grouping Genetic Algorithm (GGA)* برای خوشه‌یابی کلان‌داده‌ها ارائه شد[۱۶]. در این روش از الگوریتم ژنتیک برای نسبت دادن هر نمونه به یک خوشه‌ی مناسب استفاده شده است. برای این هدف ساختار کروموزوم‌ها به گونه‌ای طراحی شده است که هر کروموزوم شامل ژن‌هایی به تعداد نمونه‌های موجود در مجموعه داده‌ی مورد نظر می‌باشد که درون هر ژن شماره‌ی خوشه (یا برچسب) نمونه‌ی مورد نظر درج می‌شود. این ساختار کروموزوم‌ها باعث بزرگ شدن طول آن‌ها و بروز پدیده‌ی بهینه‌سازی مقیاس بزرگ[۲] و افت دقت الگوریتم به خصوص در مواجهه با کلان‌داده‌ها می‌شود. در این روش از اندیس *Silhouette* [۱۷] برای تابع برازندگی و همچنین از شاخص *Rand* برای محاسبه‌ی دقت الگوریتم ارائه شده، استفاده شده است. علی رغم جست‌وجوی فضای پاسخ توسط روش *GGA* برای یافتن تعداد خوشه‌ها، نقطه‌ی ضعف بزرگ این روش سرعت پایین آن در مواجهه با کلان‌داده‌ها می‌باشد. دلیل این امر طول بسیار زیاد کروموزوم‌ها (طول هر کروموزوم برابر است با تعداد نمونه‌های موجود در مجموعه داده) و پیچیدگی بسیار زیاد اندیس *Silhouette* است. در [۱۸] یک روش خوشه‌یابی به نام *MEPSO* با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات ارائه شده‌است. در این پژوهش از اندیس *Xie-Benni* [۱۹] به عنوان تابع برازندگی جهت ارزیابی کیفیت پاسخ‌های ارائه شده، استفاده شده است. در این پژوهش هر عامل جست‌وجو شامل دو بخش است: بخش اول دربردارنده‌ی سلول‌هایی است که تعداد آن‌ها برابر با حداکثر تعداد خوشه‌های ممکن است (c_{max}). مقدار عددی هر سلول یک و یا صفر است. یک بودن یک سلول به معنی حضور مرکز خوشه‌ی متناظر در فرایند جست‌وجوی فضای پاسخ و صفر بودن آن به معنی عدم حضور مرکز خوشه‌ی متناظر در فرایند جست‌وجو می‌باشد. بخش دوم در هر عامل جست‌وجو شامل مراکز خوشه‌ها است که طول آن برابر است با: $p \times c_{max}$. در این رابطه p تعداد ویژگی‌های داده‌ها می‌باشد. بنابراین پارامتر c_{max} باید از قبل توسط کاربر وارد شود که این نقطه ضعف بزرگی است. دقت روش پیشنهادی روی ۶ مجموعه‌داده ارزیابی شده است که از لحاظ تعداد نمونه و تعداد ابعاد از مجموعه داده‌های مصنوعیِ مورد استفاده در این پژوهش کم‌تر هستند. در یک پژوهش دیگر [۲۰] یک روش خوشه‌یابی ترکیبی با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات و روش *Kmeans* معرفی شده است. در این روش ابتدا الگوریتم ازدحام ذرات فضای پاسخ را به صورت کلی جست‌وجو کرده و سپس روش *Kmeans* یک جست‌وجوی محلی را پیرامون بهترین پاسخ یافت شده توسط الگوریتم ازدحام ذرات، انجام می دهد. در حقیقت در مرحله‌ی دوم روش *Kmeans* از خروجی الگوریتم ازدحام ذرات به عنوان مراکز خوشه‌های اولیه استفاده می‌کند. در این روش تعداد خوشه‌ها به عنوان یک پارامتر معلوم در نظر گرفته می‌شود و روش ارائه شده فضای پاسخ را برای یافتن تعداد خوشه‌ها جست‌وجو نمی‌کند. علاوه بر این استفاده از روش *Kmeans* در مرحله‌ی دوم باعث افزایش احتمال گیر افتادن در نقطه‌ی بهینه‌ی محلی می‌شود. به خصوص وقتی که تعداد نمونه‌ها و ابعاد آن‌ها زیاد باشد. در حقیقت در این روش سعی شده است با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات یکی از نقاط ضعف روش *Kmeans* پوشش داده شود اما کماکان نقاط ضعف اصلی آن که گیر افتادن در نقطه‌ی بهینه‌ی محلی و عدم توانایی در پیدا کردن تعداد خوشه‌ها است، وجود دارد. تأثیر این نقاط ضعف هنگام خوشه‌بندی کلان‌داده‌ها بیش‌تر نمایان می‌شود. در پژوهش نشان داده شده در [۲۱] یک روش خوشه‌یابی جدید بر مبنای الگوریتم ازدحام ذرات معرفی و برای تقسیم‌بندی تصویر مورد استفاده قرار گرفته است. در این روش مراکز خوشه‌ها از میان نمونه‌های موجود در مجموعه داده ی مورد نظر انتخاب می‌شوند. برای این منظور کاربر باید ابتدا حداکثر تعداد خوشه‌های ممکن (n_c) را به عنوان پارامتر ورودی وارد کند. در ابتدای فرایند، به تعداد n_c مرکز خوشه به صورت تصادفی از میان نمونه‌های موجود در مجموعه داده انتخاب می‌شود سپس برای هر پاسخ، مراکز خوشه‌ها با استفاده از شیوه‌ی کدگذاری باینری انتخاب می‌شوند. پس از اجرا شدن الگوریتم ازدحام ذرات و یافت شدن بهترین پاسخ‌ها، از الگوریتم *Kmeans* برای تصحیح کردن محل مراکز خوشه‌ها استفاده می‌شود. در مرحله‌ی بعد مجموعه‌ی مراکز خوشه‌ها که در ابتدای فرایند به صورت تصادفی انتخاب شده بودند، به‌روزرسانی می‌شود بدین صورت که مراکز یافت شده توسط روش *Kmeans* به مجموعه اضافه می‌شوند. این فرایند تا رسید به شرط خاتمه ادامه می‌یابد. همانند روش قبلی، در این روش نیز کاربر باید حداکثر تعداد خوشه‌های ممکن را به عنوان ورودی وارد کند که امری دشوار است. همچنین استفاده از روش *Kmeans* که یک روش سنتی است، با هدف بهبود موقعیت مراکز خوشه‌ها، به دلیل احتمال بالای گیرافتادن در نقطه‌ی بهینه‌ی محلی نقطه‌ی ضعف دیگر این روش می‌باشد. چرا که هیچ تضمینی برای بهبود موقعیت مراکزِ یافت شده وجود ندارد به خصوص وقتی که تعداد نمونه‌های موجود در مجموعه داده و همچنین تعداد ابعاد آن بسیار زیاد باشد. در این پژوهش اندیس‌های مختلفی برای ارزیابی کیفیت خوشه بابیِ انجام شده مورد بررسی قرار گرفته‌اند از جمله اندیس *Dunn* [۲۲]. در [۲۳] ژانگ و همکارانش یک روش خوشه‌یابی با استفاده از الگوریتم کلونی مصنوعی زنبورها[۳] ارائه کرده‌اند. در این روش برای ارزیابی کیفیت هر پاسخ، توان دوم مجموع فواصل هر نمونه تا مرکز خوشه‌ی متناظر با آن محاسبه شده است. این تابع برازندگی در قیاس با اندیس‌های دیگری که برای ارزیابی کیفیت خوشه‌یابی تاکنون معرفی شده‌اند (نظیر اندیس *Silhouette*، *Davies-Bouldin*، *Dunn* و...)، از پیچیدگی کم‌تری برخوردار است که باعث می‌شود سرعت اجرای الگوریتم افزایش یابد. اما دقت این تابع بسیار پایین است به طوری که باعث افت کیفیت پاسخ نهایی یافت شده می‌شود. در حقیقت استفاده از این تابع برازندگی می تواند نقطه‌ی ضعف این روش (به خصوص در خوشه‌یابی کلان‌داده‌ها) باشد. این روش نیز قادر به یافتن تعداد خوشه‌های موجود در مجموعه داده نمی‌باشد. علاوه بر این، روش ارائه شده تنها بر روی سه مجموعه

داده با تعداد نمونه‌ها و تعداد ویژگی‌های محدود مورد ارزیابی قرار گرفته است. بهتر بودن دقت در تنها سه مجموعه داده نسبت به سایر روش‌ها نمی‌تواند دلیل محکمی بر برتری روش ارائه شده باشد. در [۲۴] یک روش خوشه‌یابی با استفاده از الگوریتم ابتکاری هوش هیجانی[۴] ارائه شده است. روش ارائه شده در این پژوهش یک الگوریتم ترکیبی است که از ترکیب روش *Kmeans* و الگوریتم هوش هیجانی ایجاد شده است. این ترکیب همگرایی سریعتر الگوریتم هوش هیجانی را به ارمغان آورده است. با این وجود نقاط ضعف روش‌های قبلی در این روش نیز مشاهده می شود از جمله: عدم توانایی در پیدا کردن تعداد خوشه‌ها، استفاده از یک تابع برازندگی با دقت پایین (مشابه روش ارائه شده در [۲۳]) که باعث افت دقت روش پیشنهادی در مواجهه با کلان‌داده‌ها می‌شود و همچنین احتمال گیر افتادن در نقطه‌ی بهینه‌ی محلی به دلیل استفاده از روش *Kmeans*. در [۲۵] لئولیو و همکارانش، با استفاده از الگوریتم فراابتکاری[۵] جست‌وجوی تابو و روش *Kmeans* یک روش خوشه‌یابی ارائه ارائه کرده‌اند. در واقع آن‌ها با استفاده از الگوریتم جست‌وجوی تابو سعی در برطرف کردن نقاط ضعف روش *Kmeans* (گیر افتادن در نقطه‌ی بهینه‌ی محلی و وابستگی جواب نهایی به پاسخ‌های اولیه‌ی تصادفی) داشته‌اند. همچنین برای کاهش زمان اجرا، روش ارائه شده در این پژوهش با استفاده از بستر *Spark* به صورت موازی پیاده سازی شده است. این موضوع نشان می‌دهد که می‌توان الگوریتم‌های ابتکاری و فرا ابتکاری را برای پردازش‌های موازی به کار برد. نقاط ضعف این روش مانند روش‌های قبلی عدم توانایی در پیدا کردن تعداد خوشه‌ها و همچنین استفاده از تابع برازندگی غیر دقیق می‌باشد. در[۲۶] کارابوگا و همکارانش با استفاده از الگوریتم فرا ابتکاری کلونی زنبورها یک روش خوشه‌یابی ارائه کرده‌اند. آن‌ها برای سریع‌تر کردن همگرایی و همچنین جست‌وجوی بهتر فضای پاسخ، تغییراتی در الگوریتم کلونی زنبورها به وجود آورده‌اند. از جمله این که از یک تابع احتمال برای نسبت دادن نمونه‌های بدون خوشه به بهترین خوشه‌ی ممکن استفاده کرده‌اند. همچنین در مرحله‌ی دوم پس از اتمام فرایند جست‌وجو توسط الگوریتم کلونی زنبورها، روش *Kmeans* برای یافتن بهترین مراکز خوشه‌ها به کار می‌رود. به عبارت دیگر خروجی الگوریتم کلونی زنبورها به عنوان مراکز خوشه‌های اولیه در اختیار الگوریتم *Kmeans* قرار می‌گیرد. در این روش نیز تابع برازندگی، توان دوم مجموع فواصل نمونه‌های متعلق به هر خوشه تا مرکز آن خوشه می‌باشد. بزرگ‌ترین ایراد این روش نیز عدم توانایی آن در پیدا کردن تعداد خوشه‌ها می‌باشد. در پژوهش نشان داده شده در [۲۷] روش خوشه‌یابی جدید با استفاده از الگوریتم اصلاح شده‌ی جست‌وجوی گرانشی معرفی شده است. در این روش برای فرار از نقطه ی بهینه‌ی محلی و جست‌وجوی مؤثر فضای پاسخ، اصلاحاتی در الگوریتم اصلی جست‌وجوی گرانشی ایجاد شده است. بدین صورت که اگر پس از تعداد مشخصی تکرار، برازندگی بهترین پاسخ یافت شده تغییری نکرد، احتمالاً پاسخ یافت شده نقطه‌ی بهینه‌ی محلی است. برای خروج از این شرایط موقعیت هر پاسخ با میانگین گرفتن از موقعیت ۷ پاسخ نزدیک به آن اصلاح می‌شود. در این روش نیز تعداد خوشه‌ها به عنوان یک پارامتر ورودی به الگوریتم داده می‌شود در نتیجه هر عامل جست‌وجو دربردارنده‌ی تعداد k مرکز خوشه خواهد بود. بنابراین بزرگ‌ترین عیب این روش نیز عدم توانایی آن در پیدا کردن تعداد خوشه‌ها می‌باشد. همانند روش‌های قبل در این روش نیز توان دوم مجموع فواصل نمونه‌ها تا مراکز خوشه‌ی متناظر به عنوان تابع برازندگی مورد استفاده قرار گرفته است. در یک پژوهش دیگر نیز از الگوریتم فرا ابتکاری کلونی مصنوعی زنبورها و روش موازی سازی *MapReduce* برای ارائه‌ی یک روش خوشه‌یابیِ موازی استفاده شده است [۲۸]. در حقیقت این روش مشابه روش‌های پیشین است که در آن عوامل جست‌وجو به دنبال یافتن مراکز خوشه‌ها هستند. در این روش نیز تابع برازندگی، توان دوم مجموع فواصل داده‌ها از مراکز خوشه‌های متناظر است. نوآوری این روش نسبت به روش‌های دیگر تنها پیاده سازیِ موازی آن با استفاده از روش *MapReduce* و بستر *Hadoop* می‌باشد. در این روش برای افزایش سرعت اجرا، ارزیابی کیفیت پاسخ‌ها یا به عبارت دیگر محاسبه‌ی تابع برازندگی برای هر پاسخ به صورت موازی انجام می‌شود. بدین صورت که در مرحله‌ی اول (یعنی فاز *Map*) تابع *Map* برای هر عامل جست‌وجو مراکز خوشه‌ها را استخراج کرده و پس از محاسبه‌ی فواصل نمونه‌ها از مراکز خوشه‌ها، برای هر نمونه نزدیک‌ترین مرکز خوشه را مشخص می کند. این کار با استفاده از ۱۰ ماشین و به صورت موازی انجام شده است. هر ماشین این کار را بر روی بخشی از داده‌ها انجام می‌دهد. در مرحله ی دوم (یعنی فاز *Reduce*) نتایج مرحله‌ی قبل جمع بندی شده و پاسخ نهایی که همان مراکز خوشه‌ها و خوشه‌ی متناظر با هر داده است، ارائه می‌شود. با وجود موازی سازی الگوریتم ارائه شده و افزایش سرعت اجرای آن، که آن را برای کلان‌داده‌ها بسیار مفید و کاربردی می‌کند، باز هم عیب بزرگ این روش عدم توانایی در پیدا کردن تعداد خوشه‌ها می‌باشد. همان طور که بررسی شد، نقطه‌ی ضعف اکثر روش‌های اشاره شده، عدم توانایی آن‌ها در پیدا کردن تعداد خوشه‌ها است. این موضوع در مواجهه با کلان‌داده‌ها اهمیت دوچندانی پیدا می‌کند. نوآوری اصلی این پژوهش ارائه‌ی یک روش دقیق و مؤثر برای خوشه‌یابی خودکارکلان‌داده‌ها، با استفاده از یک الگوریتم بهینه‌سازی ابتکاری به نام گرگ خاکستری است.

۳- الگوریتم بهینه‌سازی گرگ خاکستری

الگوریتم بهینه‌سازی گرگ خاکستری از ساختار سلسله مراتبی جایگاه گرگ‌ها در گروه و همچنین ساختار و وظایف آن‌ها در شکار الهام گرفته شده است [۱۱]. گرگ‌های خاکستری که در بالاترین جایگاه زنجیره‌ی غذایی قرار دارند، غالباً به صورت گروهی زندگی می‌کنند. بلند مرتبه‌ترین گرگ گروه، گرگ آلفا نامیده می‌شود که مسئولیت شکار، تعیین زمان خواب، زمان بیدار شدن، مکان اسکان و سایر فعالیت‌های گروه را به عهده دارد. سایر گرگ‌ها از گرگ آلفا و تصمیمات او اطاعت می‌کنند. به عبارت دیگر گرگ آلفا عهده دار رهبری گروه است. پس از گرگ آلفا، گرگ بتا بالاترین رده را در گروه در اختیار دارد. گرگ بتا در حقیقت مشاور گرگ

آلفا در تصمیم‌گیری است اما از دستورات گرگ آلفا پیروی می‌کند. همچنین گرگ بتا کاندیدای جایگزینی گرگ آلفا در صورت پیر شدن و یا مرگ گرگ آلفا، می‌باشد. پست‌ترین جایگاه در گروه متعلق به گرگ‌های اومگا است. گاها مشاهده شده است که آن‌ها وظیفه‌ی مواظبت از توله‌ها را بر عهده دارند. گروه دیگر، گرگ‌های دلتا هستند که جایگاهی پایین‌تر از گرگ بتا و بالاتر از گرگ‌های اومگا دارند. نگهبانی، مواظبت از گرگ‌های پیر و شکار از وظایف آن‌ها است. این ساختار سلسله مراتبی و همچنین تعامل گروهی آن‌ها در شکار، بسیار قابل توجه است به نحوی که مدل سازی ریاضی این فرایند منجر به ابداع الگوریتم بهینه‌سازیِ گرگ خاکستری شده است. مراحل اصلی شکار توسط گرگ‌های خاکستری به شرح زیر می‌باشد[۲۹]:

- تعقیب و نزدیک شدن به شکار.
- محاصره و اذیت کردن طعمه تا زمان تسلیم شدن آن.
- حمله به سمت شکار.

در این الگوریتم بهینه‌سازی، عوامل جست‌وجو متناظر با گرگ‌ها، فرایند شکار متناظر با فرایند یافتن پاسخ بهینه و محل قرار گرفتن شکار متناظر با موقعیت پاسخ بهینه می‌باشد. بهترین پاسخ موجود در جمعیت متناظر با گرگ آلفا و پاسخ‌های دوم و سوم هم به ترتیب متناظر با گرگ‌های بتا و دلتا می‌باشند که سایر پاسخ‌ها جهت نزدیک شدن به پاسخ بهینه از موقعیت آن‌ها تأثیر می‌پذیرند. همان‌طور که گفته شد، گرگ‌های خاکستری در فرایند شکار پس از پیدا کردن طعمه، آن را محاصره می‌کنند. فرایند شکار معمولاً توسط گرگ آلفا هدایت می‌شود. از آن‌جایی که در فرایند جست‌وجو پاسخ‌های آلفا، بتا و دلتا بهترین و برازنده‌ترین پاسخ‌های موجود در جمعیت می‌باشند، پاسخ‌های اومگا باید موقعیت خود را با توجه به موقعیت این سه پاسخ به‌روزرسانی کنند. به عبارت دیگر در هر تکرار از الگوریتم، سه پاسخ برتر، ذخیره شده و موقعیت سایر پاسخ‌ها که پاسخ‌های اومگا نامیده می‌شوند با توجه به موقعیت این سه پاسخ به‌روزرسانی می‌شود. روابط (۲)، (۳) و (۴) نحوه ی به‌روزرسانی موقعیت پاسخ‌ها را نشان می‌دهند:

$$D_\alpha = |C_1 X_\alpha - X|, D_\beta = |C_2 X_\beta - X|, D_\delta = |C_3 X_\delta - X| \qquad (۲)$$

$$X_1 = X_\alpha - A_1 . D_\alpha, X_2 = X_\beta - A_2 . D_\beta, X_3 = X_\delta - A_3 . D_\delta \qquad (۳)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \qquad (۴)$$

در روابط فوق، D_α ، D_β و D_δ فاصله‌ی پاسخ مورد نظر به ترتیب تا پاسخ آلفا، بتا و دلتا می‌باشند. همچنین X_α، X_β و X_δ موقعیت این سه پاسخ، X موقعیت پاسخ مورد نظر، t نشان‌دهنده‌ی شماره‌ی تکرار، A و C نیز بردارهای ثابتی هستند که مقادیر آن‌ها توسط روابط زیر مشخص می شود.

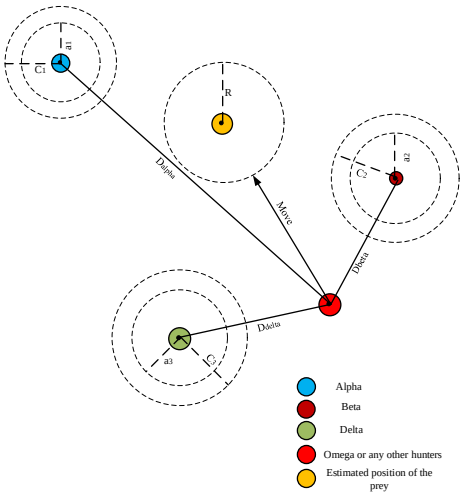
$$A = 2.a.r_1 - a \qquad (۵)$$

$$C = 2.r_2 \qquad (۶)$$

در روابط (۵) و (۶)، a برداری است که مقادیر آن بین ۲ تا صفر به صورت خطی در تکرارهای الگوریتم کاهش می‌یابد و مقادیر بردارهای r_1 و r_2 نیز اعدادی هستند که به صورت تصادفی بین صفر و یک تولید می گردند. با تغییر مقادیر a در تکرارهای مختلف طبیعتاً مقادیر المان‌های بردار A نیز در هر تکرار تغییر خواهند کرد. با توجه به رابطه‌ی (۵) مقادیر درایه‌های بردار A در بازه‌ی $[-2a \; 2a]$ قرار دارند. در روابط فوق اگر $|A| < 1$باشد، پاسخ مورد نظر به سمت موقعیت پاسخ آلفا حرکت خواهد کرد که این متناظر با همگرایی و جست‌وجو پیرامون پاسخ بهینه‌ی یافت شده (پاسخ آلفا) تا تکرار فعلی می‌باشد. برعکس اگر $|A| > 1$باشد در این صورت پاسخ‌های اومگا واگرا شده و به جست و جوی نواحی مختلف فضای پاسخ می‌پردازند. در حقیقت A یک پارامتر کنترلی است که برای کنترل فرایند جست‌وجوی فضای پاسخ و هدایت آن به سمت نواحی مختلف در نظر گرفته شده است. به عبارت دیگر این پارامتر کنترلی یک تعادل بین اکتشاف[۶] و استخراج[۷] در فرایند جست‌وجو ایجاد می‌کند. پارامتر کنترلیِ دیگر این الگوریتم، پارامتر Cاست. این پارامتر در واقع ضریبی است که میزان تأثیر موقعیت پاسخ‌های آلفا، بتا و دلتا را در نحوه ی جابه‌جاییِ پاسخ اومگای مورد نظر تعیین می‌کند. در واقع تعیین مقادیر تصادفی برای این پارامتر باعث افزایش توانایی الگوریتم مورد نظر در اکتشاف نواحی مختلف فضای پاسخ برای یافتن پاسخ بهتر و فرار از در نقطه‌ی بهینه‌ی محلی است. بدین صورت که اگر $C > 1$ حرکت پاسخ اومگا به سمت پاسخ‌های آلفا، بتا و دلتا خواهد بود و اگر $C < 1$ پاسخ مورد نظر به سمت نواحی دیگر حرکت خواهد کرد. بر خلاف A، مقادیر C در تمام تکرارها به صورت تصادفی تولید می‌شوند و این باعث افزایش خاصیت اکتشافی الگوریتم مورد نظر در جست‌وجوی فضای پاسخ نه تنها در تکرارهای اولیه بلکه در تمام تکرارها می‌شود. مراحل مختلف الگوریتم گرگ خاکستری به شرح زیر می‌باشند:

- تولید یک جمعیت اولیه‌ی تصادفی.
- محاسبه‌ی برازندگی آن‌ها.
- مشخص کردن پاسخ‌های آلفا، بتا و دلتا.
- حلقه‌ی اصلی:
  - به‌روزرسانی موقعیت همه‌ی پاسخ‌ها با استفاده از روابط (۲) تا (۶)
  - محاسبه‌ی برازندگی پاسخ‌ها
  - مشخص شدن پاسخ‌های آلفا، بتا و دلتا
  - به‌روزرسانی مقادیر پارامترهای a، A و C
- ارائه‌ی موقعیت پاسخ آلفا به عنوان بهترین پاسخ یافت شده

در تصویر شماره‌ی ۱ نحوه‌ی تغییر موقعیت پاسخ‌ها نشان داده شده است.



شکل ۱- نحوه‌ی تغییر موقعیت پاسخ‌ها [۱۱].

۴- خوشه‌یابی با استفاده از الگوریتم *GWO*

برای حل هر مساله‌ی بهینه‌سازی با استفاده از الگوریتم‌های ابتکاری ابتدا باید ساختار عوامل جست‌وجو تبیین شود و سپس در مرحله‌ی بعد یک تابع برازندگی مناسب برای عوامل جست‌وجو طراحی گردد. در مساله‌ی خوشه‌یابی ما به دنبال یافتن مراکز خوشه‌ها هستیم. در این مسیر پاسخ بهینه پاسخی است که در آن پس از نسبت دادن نمونه‌ها به نزدیک‌ترین مرکز خوشه، اصل شباهت نمونه‌های موجود در یک خوشه به هم و عدم شباهت نمونه‌های خوشه‌های مختلف به بهترین شکل ممکن رعایت شده باشد. پس عوامل جست‌وجو دربردارنده‌ی مراکز خوشه‌ها می‌باشند. در حقیقت برای یک مساله با m مرکز خوشه و n بعد، هر عامل جست‌وجو شامل $m\times n$ سلول می‌باشد. به زبان دیگر مساله‌ی یافتن m مرکز خوشه برای یک مجموعه داده با n ویژگی، یک مساله‌ی بهینه‌سازی $m\times n$ بعدی است. در شکل ۲ ساختار عوامل جست‌وجو برای یک مساله با ۴ مرکز خوشه و ۲ بردار ویژگی نشان داده شده است. در این تصویر C_{i1} و C_{i2} به ترتیب بعد اول و دوم از مرکز خوشه‌ی iام هستند. این نحوه‌ی کد گذاری عوامل جست‌وجو باعث کوتاه‌تر شدن طول کروموزوم‌ها و افزایش سرعت اجرای الگوریتم می‌شود. برای محاسبه‌ی برازندگی هر پاسخ می توان از شاخص‌های مختلفی که برای ارزیابی خوشه‌یابیِ انجام شده به کار می‌روند استفاده کرد نظیر شاخص‌های *Davies-Bouldin* [۲۲] و شاخص *Dunn* . در این پژوهش از شاخص *Calinski-Harabasz* [۳۰] برای محاسبه‌ی برازندگی پاسخ‌ها استفاده شده است.

C11	C12	C21	C22	C31	C32	C41	C42

شکل ۲- ساختار عوامل جست‌وجو برای یک مساله با ۴ مرکز خوشه و ۲ بعد

شاخص *Calinski-Harabasz* که به عنوان تابع برازندگی مورد استفاده قرار گرفته است با استفاده از روابط (۷) تا (۹) برازندگی هر پاسخ را محاسبه می‌کند:

$$VRC = \frac{SS_B}{SS_w} \times \frac{(N-k)}{k-1} \qquad (۷)$$

$$SS_B = \sum_{i=1}^{k} n_i \left\| m_i - m \right\|^2 \qquad (۸)$$

$$SS_w = \sum_{i=1}^{k} \sum_{x \in c_i} \left\| x - m_i \right\|^2 \qquad (۹)$$

در روابط بالا SS_B میزان واریانس بین خوشه‌ای[۸]، SS_W واریانس درون خوشه‌ای[۹]، k تعداد خوشه‌ها، N تعداد نمونه‌ها، m_i مرکز خوشه‌ی iام، m میانگین داده‌ها، x یک نمونه از دیتاست، c_i خوشه‌ی iام و $\|x - m_i\|$ فاصله‌ی اقلیدسی بین دو بردار می‌باشد. با توجه به روابط فوق، خوشه یابی بهتر متناظر با مقدار بزرگترِ VRC می‌باشد. پس در یک مساله‌ی کمینه‌سازی با کمینه کردن تابع $\frac{1}{VRC}$ می‌توان بهترین پاسخ ممکن را پیدا نمود. به عبارت دیگر تابع برازندگی برابر است با:

$$Fitness = \frac{1}{VRC} \qquad (۱۰)$$

در زیرقسمت‌های بعدی به ترتیب روش‌های پیشنهادی برای خوشه‌یابی غیر خودکار (حالتی که تعداد خوشه‌ها از قبل مشخص است) و خوشه یابی خودکار ارائه شده‌اند.

۴-۱- روش پیشنهادی برای خوشه‌یابی غیر خودکار کلان‌داده‌ها (*GWO-Clustering*)

در ابتدای فرایند به جای تولید پاسخ‌های کاملاً تصادفی، برای هر پاسخ (گرگ) مراکز خوشه‌ها از میان نمونه‌های موجود در مجموعه‌داده‌ی مورد بررسی و به صورت کاملاً تصادفی انتخاب می‌شوند. این کار از تولید مراکز خوشه‌های بسیار دور از محدوده‌ی داده‌ها جلوگیری می‌کند. به زبان دیگر، این نحوه‌ی تولید جمعیت اولیه باعث ممانعت از تولید پاسخ‌های اولیه‌ی بسیار دور از پاسخ بهینه‌ی کلی شده و با تولید پاسخ‌های اولیه‌ی با کیفیت‌تر، جست‌وجوی فضای پاسخ توسط الگوریتم ابتکاری به صورت موثرتری انجام خواهد شد. سپس برازندگی هر پاسخ با استفاده از روابط (۷) تا (۱۰) محاسبه می‌گردد. در این مرحله جایگاه هر پاسخ (گرگ) با توجه به مقدار برازندگی آن تعیین می‌شود. به عبارت دیگر پاسخ‌های آلفا، بتا، دلتا و اومگا مشخص می‌شوند. سپس، موقعیت پاسخ‌ها از طریق روابط (۲) تا (۴) به‌روزرسانی می‌شود. پس از هر بار به‌روزرسانی موقعیت هر پاسخ، یک مجموعه شامل تعدادی از نمونه‌های موجود در مجموعه‌داده‌ی مورد نظر که به صورت تصادفی انتخاب شده‌اند، ایجاد می‌شود. سپس در پاسخ مورد نظر مراکز خوشه‌ها به نزدیک‌ترین نمونه ی موجود در این مجموعه نسبت داده می‌شوند. یعنی با وجود این که موقعیت مراکز خوشه‌ها در هر پاسخ با استفاده از روابط (۲) تا (۴)

به‌روزرسانی می‌شود، اما در نهایت موقعیت آن‌ها موقعیت یکی از نمونه‌های موجود در مجموعه داده خواهد بود. در تصویر شماره‌ی ۳ شبه کد الگوریتم پیشنهادی نشان داده شده است.

```
Inputs : number of wolves, maxit
Output: Alpha solution
Initializing:
  For i=1 to number of wolves do:
      select k data points randomly from the data set as centroids
  End.
Calculating the fitness amount:
  For i=1 to number of wolves do:
      1. Assign each data point in the dataset to the closest centroids
      2. Calculate Calinski-Harabasz index.
  End.
Defining Alpha, Beta, Delta and Omega wolves.
Main loop:
  For it =1 to maxit do:
      For i=1 to number of population do:
          2. Calculate new positions of the Omega wolves
          3. Select j samples from the dataset randomly.
          4. Find the closest samples to the new position of the centroids
          5. Assign the position of the closest samples to the position of the
             centroids.
          6. Calculate fitness amount of the wolves.
      End.
  Update Alpha, Beta, Delta and Omega wolves..
End.
```

شکل ۳- شبه کد الگوریتم *GWO-Clustering*

۲-۴- روش پیشنهادی برای خوشه‌یابی خودکار کلان‌داده‌ها (*Automatic GWO-Clustering*)

امروزه ارائه‌ی روشی برای خوشه‌یابی خودکار که قادر به پیدا کردن تعداد خوشه‌های موجود در یک مجموعه داده باشد، بسیار حائز اهمیت است. اهمیت این موضوع با ظهور کلان‌داده‌ها و رشد جایگاه آن‌ها در بین محققان و همچنین کاربرد وسیع روش‌ها و الگوریتم‌های مرتبط با آن، افزایش می‌یابد. چرا که تشخیص تعداد خوشه‌ها و یا کلاس‌های موجود در یک مجموعه کلان‌داده کاری بسیار دشوار و تقریباً محال است. از این رو ارائه‌ی روشی که علاوه بر دقت بالا در پیدا کردن محل خوشه‌ها دقت بالایی در پیدا کردن تعداد خوشه‌ها هم داشته باشد، بسیار مورد نیاز است. در این قسمت روش ارائه شده برای خوشه‌یابی خودکار کلان‌داده‌ها که در حقیقت نوآوری اصلی این پژوهش است به صورت کامل مورد بررسی قرار می‌گیرد. روش پیشنهاد شده در زیر بخش ۴-۱ توانایی بالایی در پیدا کردن محل مراکز خوشه‌ها دارد. نتایج به دست آمده توسط این الگوریتم که در بخش بعد آورده شده‌اند، موید این امر می باشد. برای پیدا کردن تعداد خوشه‌ها یک مرحله‌ی جست‌وجوی دیگر به الگوریتم قبلی اضافه شده است. در این مرحله یک ساختار درخت گونه از الگوریتم *GWO-Clustering* اجرا می‌شود. این ساختار شامل چند درخت می‌باشد که هر درخت دربردارنده‌ی گره‌ها یا شاخه‌هایی است. هر گره درخت درواقع یک الگوریتم *GWO-Clustering* است که با تعداد خوشه‌ی مشخص اجرا می‌گردد. در حقیقت هر درخت فضای پاسخ را برای پیدا کردن تعداد مناسب خوشه‌ها جست‌وجو می‌کند. به این صورت که در ابتدای هر درخت یک جمعیت اولیه به صورتی که در زیربخش قبلی توضیح داده شد، ایجاد شده و برای تمام گره‌های درخت مورد استفاده قرار می‌گیرد. پس از ایجاد جمعیت اولیه، در گره اول یک عدد صحیح تصادفی برای k انتخاب شده و الگوریتم *GWO-Clustering* با تعداد k خوشه اجرا می‌گردد.

در پایان نتایج به دست آمده و اطلاعات مربوط به این گره شامل k و برازندگی بهترین پاسخ یافت شده در متغیری به نام حالت فعلی ذخیره می‌گردد. در ادامه در گره بعدی تعداد خوشه‌ها با استفاده از رابطه‌ی (۱۱) تغییر می‌کند.

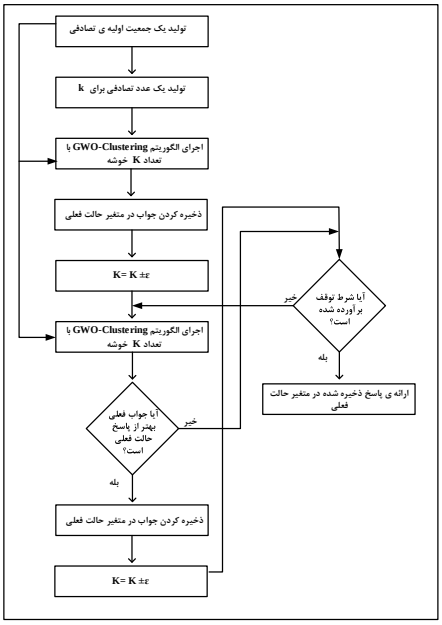
$$k_{new} = k_{old} \pm \varepsilon \qquad (۱۱)$$

در رابطه‌ی (۱۱) ε یک عدد صحیح تصادفی است. سپس در گره بعدی الگوریتم *GWO-Clustering* با تعداد جدید خوشه‌ها و جمعیت اولیه‌ی تشکیل شده در ابتدای فرایند اجرا می‌شود. در پایان نتایج این گره با حالت فعلی مقایسه می‌گردد. اگر برازندگی پاسخ جدید بهتر از برازندگی پاسخ قبلی باشد، نتایج گره جدید به عنوان حالت فعلی ذخیره می‌گردد. این روند تا انتهای گره‌های موجود در درخت ادامه می‌یابد. در شکل ۴ فلوچارت یک درخت نشان داده شده است. در روش پیشنهادی تعداد ۳ درخت اجرا شده و از میان نتایج آن‌ها بهترین نتیجه که شامل تعداد خوشه‌ها می‌باشد انتخاب می‌گردد. فاز اول الگوریتم شامل ۳ درخت بوده است که هر کدام از درخت‌ها ۱۰ گره داشته‌اند. در هرگره الگوریتم *GWO-Clustering* با تعداد ۵ پاسخ اولیه و تعداد ۱۵۰ تکرار اجرا شده است. سپس در فاز دوم که هدف یافتن موقعیت مراکز خوشه‌هاست، الگوریتم *GWO-Clustering* با تعداد ۶۰۰ تکرار برای یافتن k مرکز خوشه (خروجی فاز اول) اجرا می‌شود. در این فاز نیز جمعیت اولیه همان جمعیت اولیه‌ایست که برای بهترین پاسخ یافت شده ایجاد شده است. به عنوان مثال اگر پاسخ نهایی درخت اول بهترین پاسخ در بین سه درخت باشد، جمعیت اولیه‌ی ایجاد شده برای این درخت به عنوان جمعیت اولیه برای فاز دوم انتخاب می‌شود. ذکر این نکته نیز لازم است که بهترین پاسخ یافت شده هم به جمعیت اولیه اضافه می‌شود. هدف اصلی این مرحله یافتن موقعیت مراکز خوشه‌هاست. در شکل ۵ شبه کد الگوریتم خودکار آورده شده است. در بخش بعد مشخصات مجموعه داده‌های استفاده شده و همچنین نتایج به دست آمده برای دو حالت توضیح داده شده آورده شده‌اند.

۵- آزمایشات انجام شده

برای بررسی و ارزیابی عملکرد روش ارائه شده، ۱۳ مجموعه داده‌ی مصنوعی با ویژگی‌های مختلف انتخاب شده‌اند [۳۱]. در تمامی آزمایشات انجام شده، دقت روش پیشنهادی و دیگر روش‌های خوشه‌یابی با استفاده از شاخص *Rand* [۳۲] محاسبه شده است. همچنین در بخش مربوط به خوشه‌یابی خودکار، عملکرد روش *Xmeans* [۳۳] که قادر به جست‌وجوی فضای پاسخ برای یافتن تعداد خوشه‌ها نیز هست، مورد بررسی قرار گرفته و دقت آن با روش پیشنهادی مقایسه شده است. علاوه بر این برای انجام یک مقایسه‌ی دقیق‌تر و علمی‌تر دقت روش *Xmeans* با

استفاده از شاخص *NMI*[۱۰] [۳۴] نیز محاسبه شده و با دقت روش پیشنهادی مورد مقایسه قرار گرفته است.



شکل ۴- فلوچارت یک درخت

در پایان این بخش نیز نتایج آزمون آماری فریدمن[۱۱] [۳۵] درباره‌ی مقایسه‌ی عملکرد روش پیشنهاد شده با دیگر روش‌های خوشه‌یابی، براساس شاخص *Rand*، مورد بررسی قرار گرفته است. در زیربخش‌های ۵-۱، ۵-۲ و ۵-۳ به ترتیب مجموعه داده‌های مصنوعیِ به کار گرفته شده، شاخص *Rand* و شاخص *NMI* به طور مختصر معرفی شده‌اند. سپس در زیربخش‌های بعدی نتایج آزمایشات انجام شده آورده شده است. لازم به ذکر است که تمام آزمایشات در محیط نرم افزار متلب (نسخه‌ی ۲۰۱۶) و بر روی ماشینی با سیستم عامل windows 10 و مشخصات زیر پیاده سازی شده‌اند:

CPU: Core i7-4700 MQ, 2.4 GHz, RAM: 8GB

۵-۱- معرفی مجموعه داده‌های مصنوعی

مجموعه داده‌های *S*: این مجموعه داده‌های دو بعدی (*S1*، *S2*، *S3* و *S4*) دارای ۵۰۰۰ نمونه در ۱۵ خوشه‌ی گوسی با درجات همپوشانی متفاوت هستند به طوری که از *S1* تا *S4* درجه‌ی همپوشانی خوشه‌ها افزایش می‌یابد. یعنی در *S4* میزان همپوشانی خوشه‌ها بیش از سه مجموعه‌ی دیگر است [۳۶].

مجموعه داده‌های *A*: مجموعه داده‌های *A1*، *A2* و *A3* به ترتیب دارای ۲۰، ۳۵ و ۵۰ خوشه‌ی دایره‌ای هستند که هر خوشه دارای ۱۵۰ نمونه ی دو بعدی می‌باشد. میزان همپوشانیِ خوشه‌ها کمتر از مجموعه داده‌های *S* می‌باشد [۳۷].

مجموعه داده‌های *G2*: مجموعه داده‌های *G2-32-60*، *G2-128-60*، *G2-256-60* و *G2-1024-70* که همگی دارای ۲۰۴۸ نمونه در دو خوشه‌ی گوسی می‌باشند به ترتیب دارای۳۲، ۱۲۸، ۲۵۶ و ۱۰۲۴ بردار ویژگی هستند. همچنین انحراف معیار داده‌ها برای سه مجموعه داده‌ی اول ۶۰ و برای مجموعه داده‌ی *G2-1024-70*، برابر با ۷۰ می‌باشد [۳۸].

مجموعه داده‌های *High dimensional*: این مجموعه داده‌ها دارای ۱۰۲۴ نمونه در ۱۶ خوشه‌ی کاملاً متمایز می‌باشند. مجموعه داده‌ی *Dim032* دارای ۳۲ بردار ویژگی و مجموعه داده‌ی *Dim064* دارای ۶۴ بردار ویژگی می‌باشند [۳۹]. در جدول شماره‌ی ۱ مشخصات کلی این مجموعه داده‌ها آورده شده است.

۵-۲- شاخص *Rand*

این شاخص برای اندازه‌گیری دقت خوشه‌یابیِ انجام شده در حالتی که خوشه‌ی داده‌ها از قبل مشخص است، استفاده می‌شود. در حقیقت این شاخص برای ارزیابیِ عملکرد الگوریتم‌های خوشه‌یابی و مقایسه‌ی آن‌ها مورد استفاده قرار می‌گیرد. شاخص *Rand* درصد تصمیمات درست اتخاذ شده توسط الگوریتم خوشه‌یابی را، درنسبت دادن نمونه‌های مختلف به خوشه‌های مختلف، محاسبه می‌کند. برای یک مجموعه داده با n نمونه اگر $y = \{y_1, y_2, \dots, y_n\}$ مجموعه‌ی برچسب‌های داده‌ها باشد و $x = \{x_1, x_2, \dots x_n\}$برچسب‌های به دست آمده توسط الگوریتم باشد، شاخص *Rand* با استفاده از رابطه‌ی ۱۲ محاسبه می‌گردد:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \qquad (۱۲)$$

در رابطه‌ی فوق :

a: تعداد جفت نمونه‌هایی که هم در مجموعه‌ی x و هم در مجموعه‌ی y در یک خوشه قرار دارند.

b: تعداد جفت نمونه‌هایی که هم در مجموعه‌ی x و هم در مجموعه‌ی y در دو خوشه‌ی متفاوت قرار دارند.

c: تعداد جفت نمونه‌هایی که در مجموعه‌ی x در یک خوشه قرار دارند اما در مجموعه‌ی y در دو خوشه‌ی متفاوت واقع شده‌اند.

جدول ۱- مشخصات مجموعه داده‌های بررسی شده

	مجموعه داده	تعداد نمونه‌ها	تعداد ویژگی‌ها	تعداد خوشه‌ها
S datasets	*S1* *S2* *S3* *S4*	۵۰۰۰	۲	۱۵
A datasets	*A1*	۳۰۰۰	۲	۲۰
	A2	۵۲۵۰	۲	۳۵
	A3	۷۵۰۰	۲	۵۰
G2 datasets	*G2-32-60*	۲۰۴۸	۳۲	۲
	G2-128-60	۲۰۴۸	۱۲۸	۲
	G2-256-60	۲۰۴۸	۲۵۶	۲
	G2-1024-70	۲۰۴۸	۱۰۲۴	۲
High dimensional datasets	*Dim032*	۱۰۲۴	۳۲	۱۶
	Dim064	۱۰۲۴	۶۴	۱۶

d: تعداد جفت نمونه‌هایی که در مجموعه‌ی x در دو خوشه‌ی متفاوت قرار دارند اما در مجموعه‌ی y در یک خوشه قرار دارند.

۵-۳- شاخص *NMI*

در بعضی موارد، الگوریتم خوشه‌یابیِ مورد نظر تعداد خوشه‌ها را به درستی پیدا می‌کند اما بعد از نسبت دادن هر نمونه به نزدیک‌ترین مرکز خوشه، مشاهده می‌شود که شکل خوشه‌ها با واقعیت فاصله دارد. به عبارت دیگر دقت بالا در پیدا کردن تعداد خوشه‌ها لزوماً به معنی دقت بالای خوشه‌یابی نیست. برای بررسی دقیق شباهت خوشه‌های یافت شده به خوشه‌های اصلی از شاخص *NMI* استفاده می‌شود که این شاخص از طریق رابطه‌ی زیر میزان شباهت بین دو خوشه را محاسبه می‌کند:

$$NMI(Y,C)=\frac{2\times I(Y;C)}{H(Y)+H(C)} \qquad (۱۳)$$

در رابطه‌ی شماره‌ی ۱۳، Y شماره‌ی کلاس نمونه‌ها، C برچسب خوشه ی آن‌ها که توسط الگوریتم خوشه‌یابی مشخص شده است، H تابع آنتروپی و I تابع مشخص کننده‌ی اطلاعات مشترک می‌باشد. این دو تابع به صورت زیر محاسبه می‌شوند:

$$H=-\sum_{i=1}^{n} p_i \log p_i \qquad (۱۴)$$

$$I(x,y)=\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x).p(y)} \qquad (۱۵)$$

هرچه مقدار *NMI* بیش‌تر باشد، خوشه‌های یافت شده بیش‌تر شبیه خوشه‌های اصلی هستند.

۵-۴- نتایج به دست آمده برای خوشه‌یابی غیر خودکار

در جداول ۲ تا ۵ نتایج حاصل از اعمال روش پیشنهادی بر روی مجموعه داده‌های مصنوعیِ معرفی شده در زیربخش ۵-۱، قابل مشاهده است. برای ارزیابی دقت روش پیشنهادی شاخص *Rand* به کار برده شده است. در این جدول‌ها عملکرد الگوریتم *GWO-Clustering* با روش *Kmeans* ، روش *PSO* و یک الگوریتم خوشه‌یابی فازی مبتنی بر وزن دهی به ویژگی‌ها به نام *RLFWFCM* [۴۰] مورد مقایسه قرار گرفته است. جداول ۲ تا ۵ برتری روش پیشنهادی را بر روش *Kmeans* که یکی از متداول‌ترین روش‌های خوشه‌یابی می‌باشد نشان می‌دهند. به زبان ساده‌تر روش *GWO-Clustering* دقت بالایی در پیدا کردن مراکز خوشه‌ها دارد. در تصاویر ۶ تا ۹ نیز این موضوع به خوبی نشان داده شده است. در این تصاویر، نقاط سیاهرنگ مراکز خوشه‌ها هستند که توسط الگوریتم یافت شده‌اند. همچنین نقاط آبی رنگ مشخص کننده‌ی نمونه‌های موجود در مجموعه داده‌ی مورد بررسی می‌باشند.

تصاویر ۶ تا ۹ نشان‌دهنده‌ی دقت بالای روش پیشنهادی در پیدا کردن مراکز خوشه‌ها برای مجموعه‌داده‌های با تعداد خوشه‌های زیاد و حتی تعداد ابعاد زیاد می‌باشند. دقت صد در صدی به دست آمده برای مجموعه‌داده‌های *g2-1024-70* و *dim064* که هر کدام به ترتیب دارای ۱۰۲۴ و ۶۴ بردار ویژگی و همچنین ۲ و ۱۶ خوشه می‌باشند نشان از توان بالای روش خوشه‌یابی ارائه شده می‌باشد.

```
Stage 1:
Inputs: number of trees, length of tree, number of population
For i=1 to number of trees do:
    1.k = Generate a random integer number
    2. beginning population = Generate a random population
    3.Current state.fitness = inf
    4. Current state.k=Ø
    for j=1 to length of tree do:
        if (j==1) do:
            current state.k=k
        end
        best fitness = GWO-Clustering(k , beginning population)
        if best fitness < current state.fitness
            current state.fitness=best fitness
            curent state.k=k
        end
        k = current state.k ± ε
    end
Bestk[i] = current state.k
End
Output: find the best solution which has the best fitness amount and its corresponding k
and its corresponding beginning population
Stage 2:
Final output = GWO-Clustering(best k, beginning population)
```

شکل ۵- شبه کد الگوریتم خودکار خوشه‌یابی کلان‌داده‌ها

جدول ۲- نتایج به دست آمده برای مجموعه داده‌های *S*

روش	*S1*	*S2*	*S3*	*S4*
GWO-Clustering	**۰/۹۹۸۴**	**۰/۹۹۲۳**	**۰/۹۶۵۷**	**۰/۹۵۴۸**
Standard PSO	۰/۹۸۸۳	۰/۹۸۵۰	۰/۹۵۵۷	۰/۹۵۰۳
Kmeans	۰/۹۹۰۱	۰/۹۷۷۷	۰/۹۵۲۲	۰/۹۴۷۲
RLFWFCM	۰/۹۹۲۷	۰/۹۹۲۳	۰/۹۶۲	۰/۹۵۱۶

جدول ۳- نتایج به دست آمده برای مجموعه داده‌های *A*

روش	*A1*	*A2*	*A3*
GWO-Clustering	**۰/۹۹۹۳**	**۰/۹۹۹۳**	۰/۹۸۷۳
Standard PSO	۰/۹۶۱۷	۰/۹۸۱۱	**۰/۹۸۸۹**
Kmeans	۰/۹۸۷۷	۰/۹۹۲۴	۰/۹۹۴۹
RLFWFCM	۰/۹۹۱۸	۰/۹۹۴۶	۰/۹۹۱۷

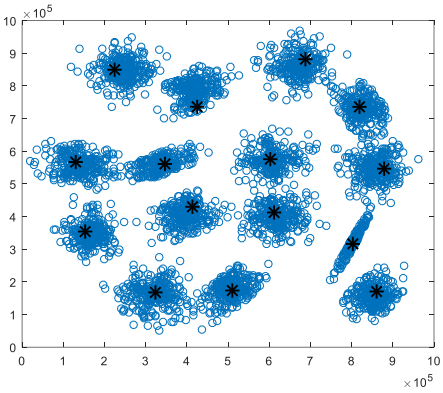
جدول ۴- نتایج به دست آمده برای مجموعه داده‌های *G2*

روش	*G2-32-60*	*G2-128-60*	*G2-256-60*	*G2-1024-70*
GWO-Clustering	**۱**	**۱**	**۱**	**۱**
Standard PSO	۰/۸۵۱۷	۰/۹۴۲۲	۰/۸۹۹۸	۰/۹۰۹۵
Kmeans	۱	۱	۱	۱
RLFWFCM	۱	۱	۱	۱

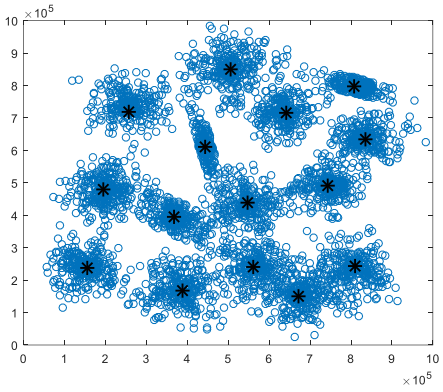
جدول ۵- نتایج به دست آمده برای مجموعه داده‌های *high dimensional*

روش	*Dim032*	*Dim064*
GWO-Clustering	**۱**	**۱**

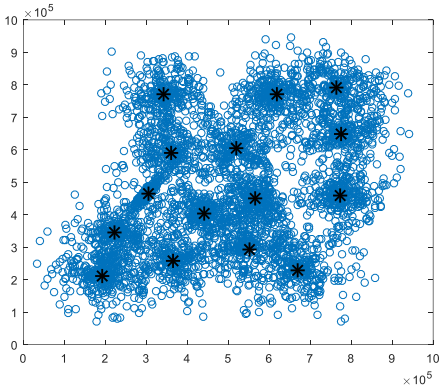
Standard PSO	۰/۹۵۷۶	۰/۹۶۶
Kmeans	۱	۱
RLFWFCM	۰/۵۵۸۱	۰/۶۹۶۵



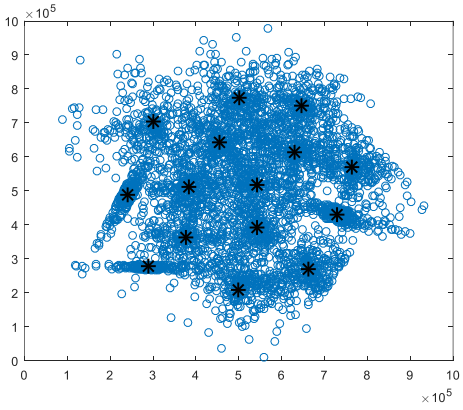
الف



ب

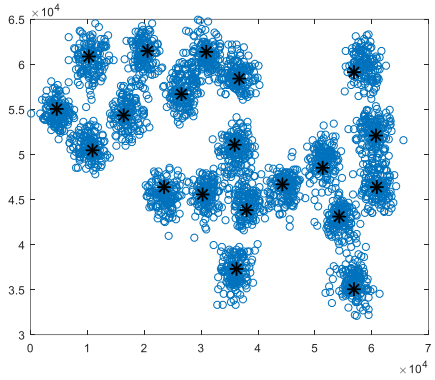


ج

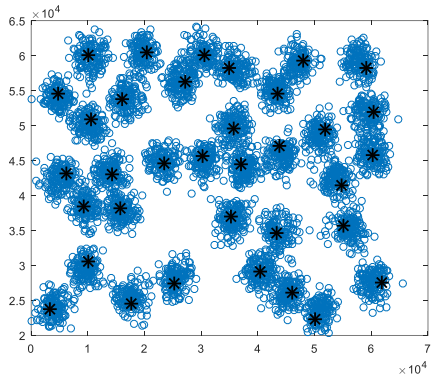


د

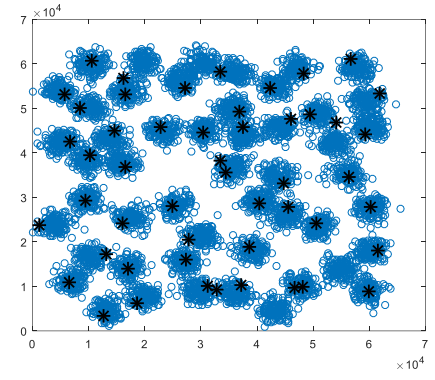
شکل ۶ الف، ب، ج، د- مراکز خوشه‌های به دست آمده توسط روش پیشنهادی به ترتیب برای مجموعه داده‌ی S1، S2، S3 و S4



الف

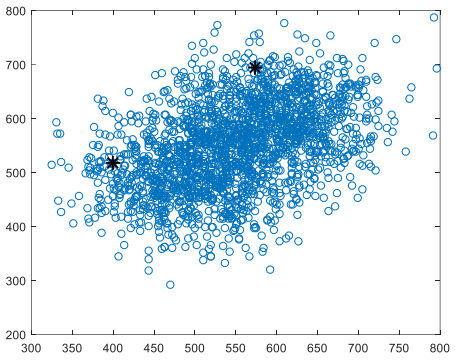


ب

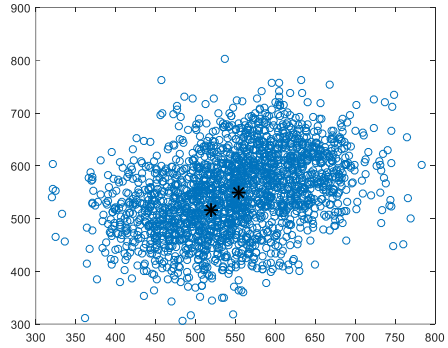


ج

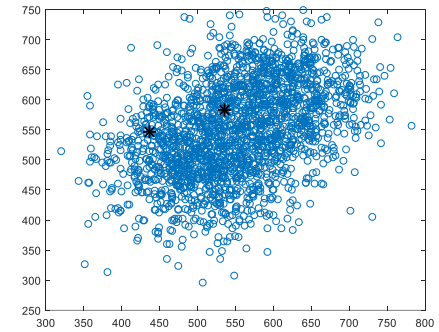
شکل ۷ الف، ب، ج- مراکز خوشه‌های به دست آمده توسط روش پیشنهادی به ترتیب برای مجموعه داده‌های A1، A2 و A3



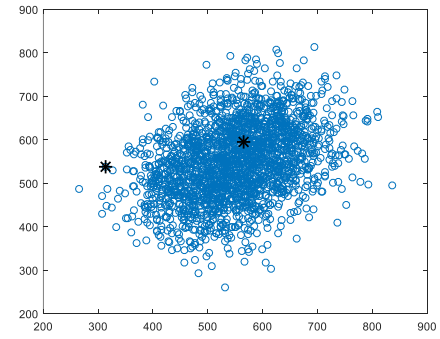
الف



ب

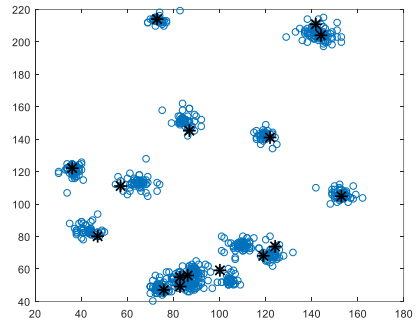


ج

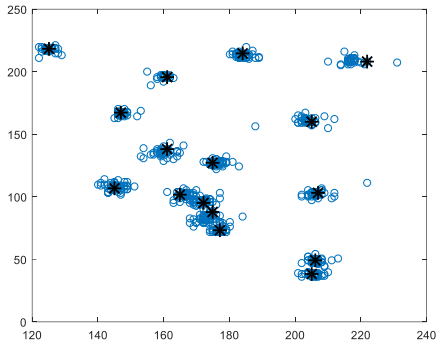


د

شکل ۸ الف، ب، ج، د- مراکز خوشه‌های به دست آمده توسط روش پیشنهادی به ترتیب برای مجموعه داده‌های g2-32-60، g2-128-60، g2-256-60 و g2-1024-70



الف



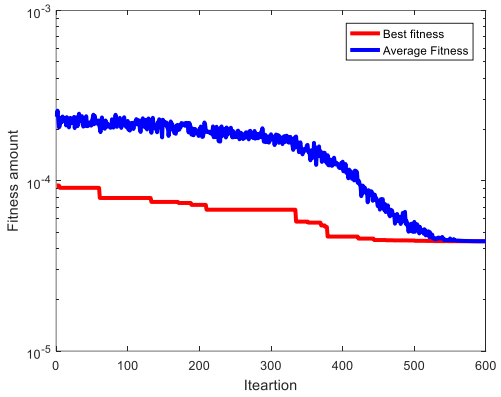
ب

شکل ۹ الف و ب- مراکز خوشه‌های به دست آمده توسط روش پیشنهادی به ترتیب برای مجموعه داده‌های dim032 و dim064

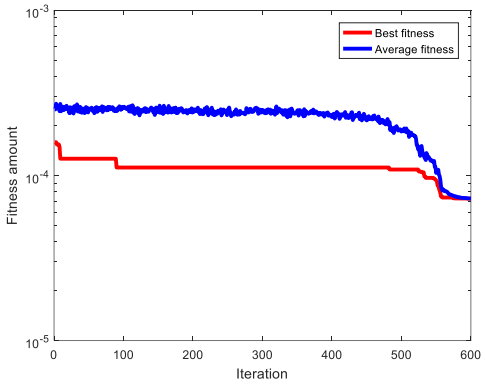
علاوه بر این با مقایسه‌ی نمودارهای همگرایی که در تصایر ۱۰ و ۱۱ به ترتیب برای الگوریتم *GWO-Clustering* و *PSO* آورده شده‌اند، به سادگی می‌توان جست‌وجوی مؤثر فضای پاسخ را توسط الگوریتم *GWO-Clustering* مشاهده نمود. با توجه به تصویر ۱۱، الگوریتم *PSO* در همان تکرارهای اولیه به جواب نهایی همگرا شده است. این موضوع می‌تواند باعث گیر افتادن در نقطه‌ی بهینه‌ی محلی شود در حالی که الگوریتم *GWO-Clustering* پس از جست‌وجوی مؤثر فضای پاسخ در طی تکرارهای مختلف، در تکرارهای آخر همگرا شده و به جست‌وجوی نواحی پیرامون نقطه‌ی بهینه پرداخته است. در تمام آزمایشات انجام شده، تعداد خوشه‌ها به عنوان یک پارامتر معلوم در نظر گرفته شده است در حالی که در بسیاری از مسائل خوشه‌یابی، به خصوص در کلان‌داده‌ها، اطلاعات کافی در مورد مجموعه داده‌ی مورد نظر و تعداد خوشه‌های آن در دسترس نیست. از این رو پیدا کردن تعداد خوشه‌ها چالشی بسیار مهم می‌باشد. در زیربخش بعدی نتایج قابل توجه به دست آمده توسط الگوریتم خوشه‌یابی خودکار که قادر به یافتن تعداد خوشه‌ها نیز می باشد، ارائه شده است.

۵-۵- نتایج به دست آمده برای خوشه‌یابی خودکار (*Automatic GWO-Clustering*)

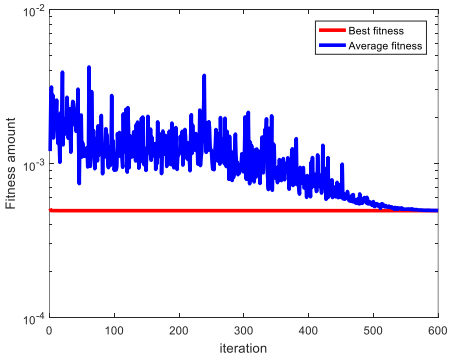
میانگین نتایج به دست آمده توسط الگوریتم خوشه‌یابی خودکار و روش *Xmeans* در جداول ۶ تا ۱۳ آورده شده است. نتایج مندرج در این جداول نشان از دقت بالای الگوریتم پیشنهادی در پیدا کردن تعداد خوشه‌ها دارد. با توجه به این جداول می‌توان به وضوح برتری روش پیشنهادی را بر روش *Xmeans* مشاهده کرد. از منظر هر دو معیار *Rand* و *NMI* روش *AGWO-Clustering* عملکرد بهتری داشته است. این روش در قیاس با روش *Xmeans*، که یک روش خوشه‌یابی خودکار است، نه تنها با دقت فوق العاده‌ای تعداد خوشه‌ها و مراکز آن‌ها را پیدا کرده است، بلکه زمان اجرای آن نیز بسیار کم‌تر از روش *Xmeans* می‌باشد.



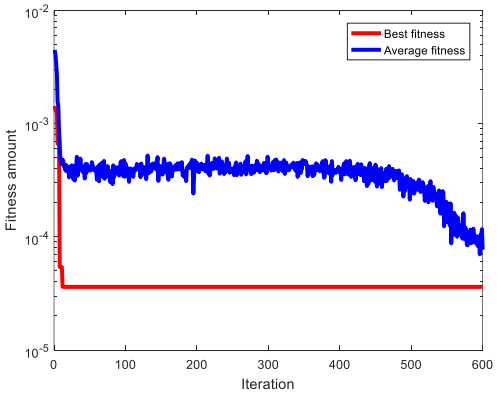
الف



ب

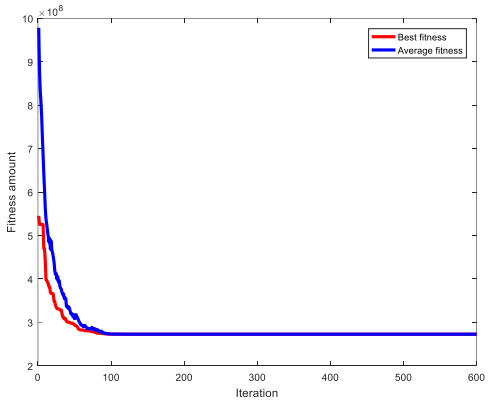


ج

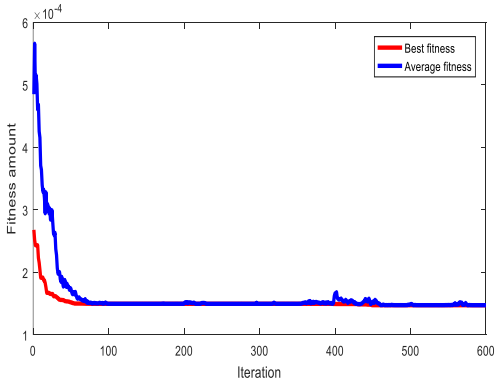


د

شکل ۱۰ الف، ب، ج، د- نمودارهای همگرایی الگوریتم پیشنهادی به ترتیب برای مجموعه داده‌های S1، A1، g2-32-60 و dim032



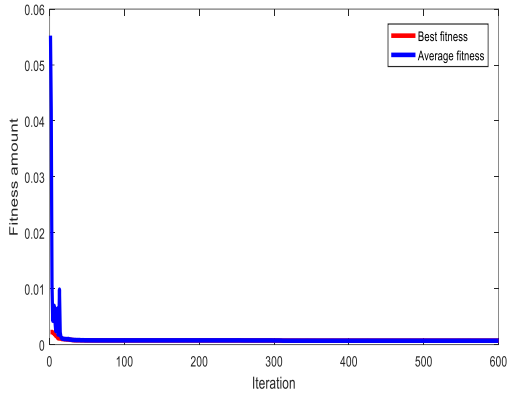
الف



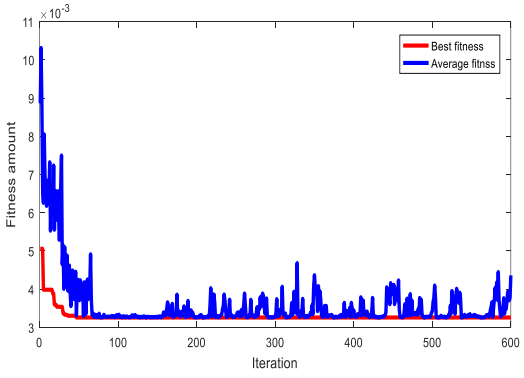
ب

جدول ۹- میانگین نتایج به دست آمده توسط روش *Xmeans* برای *AGWO-Clustering* برای مجموعه داده‌های *A*

مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
S1	۰/۹۸۶۱۴	۰/۸۰۹۴	۲۰	۶	۶۶۰۱۶۵/۸۸
S2	۰/۹۹۴۲۶	۰/۹۲۴	۳۵	۹۳۸	۷۶۹۴۹/۳۵
S3	۰/۹۹۱۲۷	۰/۸۱۸۹	۵۰	۹۵۵	۱۸۳۴۵۰/۰۴
S4	۰/۹۱۹۹	۰/۶۶۶	۱۵	۱۰	۶۴۷۲۳/۹۲



ج



د

شکل ۱۱ الف، ب، ج، د- نمودارهای همگرایی الگوریتم standard PSO به ترتیب برای مجموعه داده‌های S1، A1، g2-32-60 و dim032

جدول ۶- میانگین نتایج به دست آمده توسط روش *AGWO-Clustering* برای مجموعه داده‌های *S*

مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
S1	۰/۹۹۵۹	۰/۹۸۵۵	۱۵	۱۶	۷۵۳/۱۵
S2	۰/۹۹۲۳	۰/۹۳۹۴	۱۵	۱۶/۵	۷۶۴/۳۳
S3	۰/۹۶۶	۰/۷۸۷۸	۱۵	۱۴/۷۵	۸۲۸/۳۶
S4	۰/۹۵۴	۰/۷۲	۱۵	۱۴/۷۵	۷۷۱/۸۴

جدول ۱۰- میانگین نتایج به دست آمده توسط روش *AGWO-Clustering* برای مجموعه داده‌های *G2*

مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
G2-32-60	۱	۱	۲	۲	۵۱۸/۷۶
G2-128-60	۱	۱	۲	۲	۶۸۰/۷۲
G2-256-60	۱	۱	۲	۲	۹۲۶/۴۶
G2-1024-70	۱	۱	۲	۲	۲۸۸۶

جدول ۱۱- میانگین نتایج به دست آمده توسط روش *Xmeans* برای مجموعه داده‌های *G2*

مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
G2-32-60	۰/۵۰	۰/۳۱۳	۲	۱۰	۱۰۱۳۹/۵۹
G2-128-60	۰/۵۰	۰/۳۰۹	۲	۱۱	۱۶۶۸۰/۰۱
G2-256-60	۰/۵۰	۰/۳۰۸	۲	۱۵	۲۲۷۱۵/۱۲
G2-1024-70	۰/۵۰	۰/۳۰۷	۲	۱۷	۷۰۶۴۴/۲۱

جدول ۱۲- میانگین نتایج به دست آمده توسط روش *AGWO-Clustering* برای مجموعه داده‌های *High dimensional*

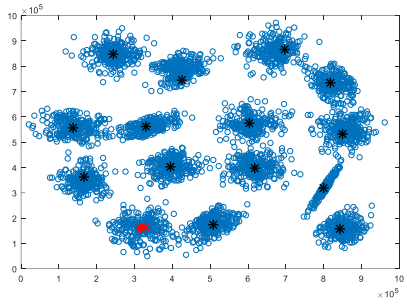
مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
Dim032	۱	۱	۱۶	۱۶	۴۶۵/۰۹
Dim064	۱	۱	۱۶	۱۶	۵۱۸/۶۳

جدول ۱۳- میانگین نتایج به دست آمده توسط روش *Xmeans* برای مجموعه داده‌های *High dimensional*

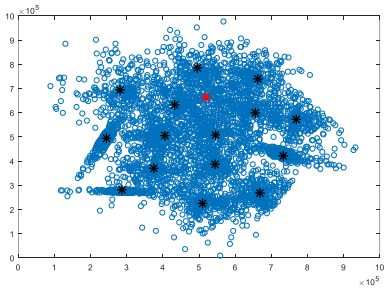
مجموعه داده	دقت		تعداد خوشه‌ها	تعداد خوشه‌های یافت شده	زمان اجرا (برحسب ثانیه)
	Rand	*NMI*			
Dim032	۰/۹۸۴	۰/۹۶۸	۱۶	۱۴	۱۸۹۹/۷
Dim064	۰/۹۸۴	۰/۹۶۸	۱۶	۱۴	۲۱۴۷/۸

این موضوع در خوشه‌یابی کلان‌داده‌ها اهمیت فوق العاده‌ای دارد. همچنین این جداول پیچیدگی بحث خوشه‌یابی خودکار و جایگاه روش‌های ابتکاری و فرا ابتکاری در خوشه‌یابی خودکار را نشان می‌دهند. جست‌وجوی فضای پاسخِ پیچیده که ناشی از تعداد زیاد نمونه‌ها و ابعاد زیاد آن‌ها است، امری دشوار و زمان‌بر است که الگوریتم‌های سنتی خوشه‌یابی توانایی انجام آن را ندارند. روش پیشنهاد شده در این پژوهش به خوبی این چالش را مرتفع کرده است. نتایج مندرج در جداول ۶ تا ۱۳ موید این موضوع هستند. آن‌چه بسیار قابل توجه است، عملکرد خوب روش پیشنهادی در مورد مجموعه داده‌های با تعداد خوشه‌ها و همچنین با تعداد ابعاد زیاد می‌باشد.

علاوه بر این حتی در مورد مجموعه داده‌هایی که تعداد خوشه‌ها به صورت کاملاً دقیق پیدا نشده است، باز هم مشاهده می‌شود که دقت به دست آمده حتی از روش الگوریتم *PSO* و همچنین روش متداول *Kmeans* که روش‌های غیرخودکار هستند و نتایج آن‌ها در بخش قبل آورده شده، بهتر است. به عبارت دیگر روش ارائه شده نه تنها دقت بالایی در پیدا کردن موقعیت مراکز خوشه‌ها دارد بلکه دقت فوق العاده‌ای در پیدا کردن تعداد خوشه‌ها ارائه کرده است. در تصاویر شماره‌ی ۱۲ و ۱۳ مراکز خوشه‌های پیدا شده توسط الگوریتم پیشنهادی برای مجموعه داده‌های *S1*، *S4* ، *A1*، *A2* و *A3* نشان داده شده‌اند.

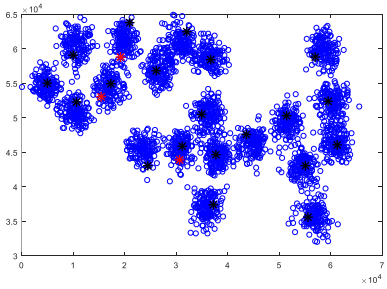


الف

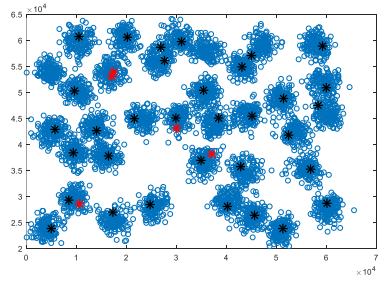


ب

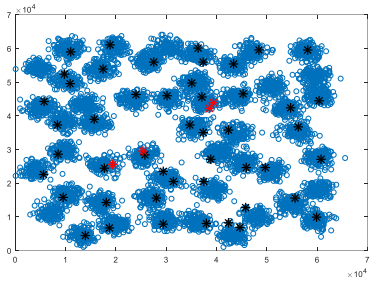
شکل ۱۲ الف و ب- مراکز خوشه‌های یافت شده توسط AGWO-Clustering به ترتیب برای S1 و S4



الف

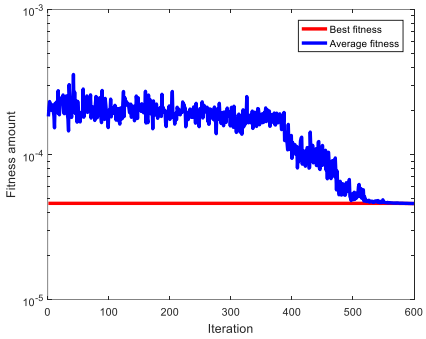


ب

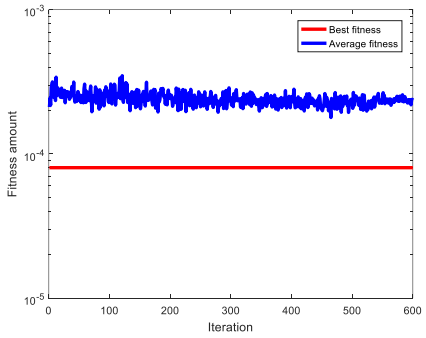


ج

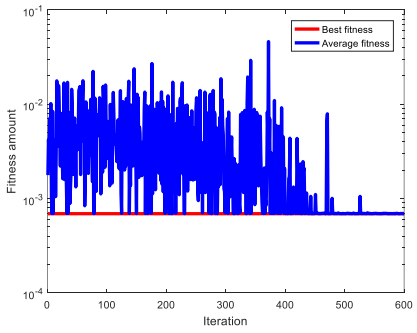
شکل ۱۳ الف، ب، ج- مراکز خوشه‌های یافت شده توسط الگوریتم AGWO-Clustering به ترتیب برای A1، A2 و A3



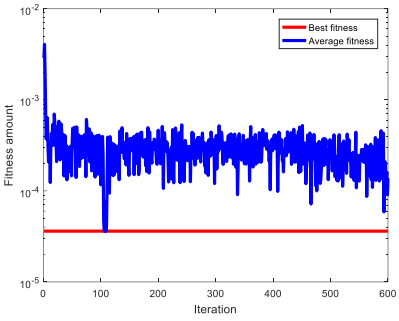
الف



ب



ج



د

شکل ۱۴ الف، ب، ج، د- نمودارهای همگرایی الگوریتم AGWO-Clustering به ترتیب برای مجموعه داده‌های S1، A1، g2-32-60 و dim032

این تصاویر خروجی الگوریتم در یک آزمایش خاص را نشان می‌دهند. آن‌چه از تصاویر ۱۲ و ۱۳ کاملاً مشهود می‌باشد این است که برای مجموعه داده‌هایی که تعداد خوشه‌ها با دقت کامل یافت نشده است، مراکز خوشه‌های اضافیِ یافت شده توسط الگوریتم که در تصاویر با رنگ قرمز مشخص شده‌اند، بسیار نزدیک به سایر مراکز می‌باشند به طوری که در اکثر موارد می‌توان این خوشه‌های اضافی را با خوشه‌های اصلی ادغام کرد. در هر صورت حتی با وجود خوشه‌های اضافی باز هم دقت روش پیشنهادی قابل قبول و در اکثر آزمایشات بهتر از سایر روش‌ها می باشد. در شکل ۱۴ نیز نمودارهای همگرایی روش خوشه‌یابی خودکار برای چهار مجموعه داده‌ی نمونه و برای آزمایش‌های خاص آورده شده است. نمودارهای همگراییِ به دست آمده همانند حالت قبل نشان از جست‌وجوی نواحی مختلف فضای پاسخ توسط الگوریتم گرگ خاکستری دارند. برخلاف الگوریتم ازدحام ذرات که اعضای جمعیت در همان تکرارهای اولیه به بهترین پاسخ یافت شده همگرا می‌شوند، در الگوریتم گرگ خاکستری، پاسخ‌ها پس از جست‌وجوی مؤثر بخش‌های مختلف فضای پاسخ در تکرارهای آخر همگرا می‌شوند.

این نکته احتمال گیر افتادن در نقاط بهینه‌ی محلی را کمتر می کند. در کنار دقت بالا آنچه در پردازش کلان‌داده‌ها حائز اهمیت است، زمان اجرای الگوریتم‌های مورد استفاده می‌باشد. به زبان ساده‌تر استخراج اطلاعات مفید از کلان‌داده‌ها حتی با دقت بالا در یک بازه‌ی زمانی بسیار زیاد قابل قبول نیست. با وجود این که روش پیشنهادی بسیار سریع‌تر از روش خودکار *Xmeans* است اما این به معنی کامل بودن آن نمی‌باشد. روش‌ها و تکنیک‌های متعددی برای کاهش زمان اجرا و همچنین افزایش سرعت پردازش کلان‌داده‌ها معرفی شده‌اند از جمله تکنولوژی‌ها و تکنیک‌های مربوط به موازی سازی روش‌های مرتبط با کلان‌داده‌ها نظیر *SPARK* و *MapReduce*. تاکنون پژوهش‌های زیادی برای پیاده‌سازی روش‌های هوش جمعی به صورت موازی در زمینه‌های مختلف انجام شده است. به عنوان مثال در [۴۱] یک روش خوشه‌یابیِ موازی با استفاده از الگوریتم ازدحام ذرات و روش *MapReduce* ارائه شده است. در این پژوهش الگوریتم ازدحام ذرات با استفاده از سه ماژولِ *MapReduce* پیاده سازی گردیده است.

دو ماژول اول به ترتیب برای به‌روزرسانی موقعیت اعضای جمعیت و به‌روزرسانی مقدار برازندگی آن‌ها و ماژول سوم برای ادغام نتایج حاصل از دو ماژول اول در نظر گرفته شده‌اند.

همچنین پژوهش‌های متعددی با استفاده از الگوریتم‌های موازی هوش جمعی تاکنون انجام شده است[۴۴-۴۲]. بنابراین با توجه به امکان موازی سازی الگوریتم‌های ابتکاری بخصوص روش‌های هوش جمعی، امکان موازی سازی الگوریتم خوشه‌یابی خودکارِ ارائه شده در این پژوهش نیز وجود دارد که این باعث افزایش بهره‌وری این روش وهمچنین افزایش مقیاس پذیری آن می‌شود به طوری که با زیادتر شدن نمونه‌های یک مجموعه داده باز هم روش ارائه شده قابلیت اجرایی داشته باشد.

۶-۵- نتایج تحلیل آماری فریدمن

تحیل آماری فریدمن معادل غیر پارامتری آزمون *ANOVA* است [۳۵]. بر اساس این تحلیل آماری، الگوریتم‌های خوشه‌یابی براساس عملکردشان بر روی مجموعه داده‌های مختلف رتبه‌بندی می‌شوند. این کار در جدول شماره‌ی ۱۴ انجام شده است. به عنوان مثال، با توجه به جداول بخش قبل، عملکرد الگوریتم *AGWO-Clustering* بر روی مجموعه داده‌ی *S1* بهتر از بقیه‌ی روش‌ها است.

جدول ۱۴- رتبه بندی الگوریتم‌ها بر اساس عملکردشان بر روی مجموعه داده‌های مختلف بر پایه‌ی شاخص *Rand*

	AGWO-Clustering	*PSO*	*Kmeans*	*RLFWFCM*	*Xmeans*
S1	۱	۴	۳	۲	۵
S2	۱/۵	۳	۴	۱/۵	۵
S3	۱	۳	۴	۲	۵
S4	۱	۳	۴	۲	۵
A1	۱	۴	۳	۲	۵
A2	۲	۴	۳	۱	۵
A3	۲/۵	۴	۱	۲/۵	۵
G2-32-60	۲	۴	۲	۲	۵
G2-128-60	۲	۴	۲	۲	۵
G2-256-60	۲	۴	۲	۲	۵
G2-1024-70	۲	۴	۲	۲	۵
Dim032	۱/۵	۴	۱/۵	۵	۳
Dim064	۱/۵	۴	۱/۵	۵	۳
مجموع	۲۱	۴۹	۳۳	۳۱	۶۱

در نتیجه رتبه‌ی روش *AGWO-Clustering* برای این مجموعه داده برابر با ۱ خواهد بود. پس از رتبه‌بندی الگوریتم‌ها، جمع رتبه‌های کسب شده توسط هر الگوریتم بر روی مجموعه داده‌های مختلف محاسبه شده و پس از آن مقدار آماری زیر به دست می‌آید:

$$F=\left(\frac{12}{\left[N\times K\times(K+1)\right]}\right)\times\sum R^{2}-\left[3\times N\times(K+1)\right] \qquad (۱۶)$$

توضیح پارامترهای به کار رفته در رابطه‌ی فوق به شرح زیر می‌باشد :

N: تعداد مجموعه داده‌ها

K: تعداد الگوریتم‌ها

R: جمع رتبه‌های به دست آمده برای هر الگوریتم

بزرگ‌تر بودن مقدار به دست آمده توسط این رابطه از مقدار بحرانیِ مرتبط با مساله‌ی مورد نظر، به معنی وجود تفاوت قابل توجه در عملکرد روش‌های ارائه شده می‌باشد. در غیر این صورت عملکرد روش‌ها نزدیک به هم می‌باشد. نکته‌ی قابل ذکر در مورد این روش این است که اگر عملکرد دو روش در یک مجموعه داده یکسان باشد رتبه‌ی آن‌ها میانگین گرفته می‌شود. جدول شماره‌ی ۱۴ اطلاعات مربوط به تحلیل آماری فریدمن را نشان می‌دهد. از جدول فوق به وضوح می‌توان برتری روش پیشنهادی را مشاهده کرد. باید به این نکته نیز توجه داشت که روش پیشنهادی روش خوشه‌یابی خودکار است که علاوه بر موقعیت مراکز خوشه‌ها تعداد خوشه‌ها را نیز پیدا می‌کند. در حالی که سه روش *PSO*، *Kmeans* و *RLFWFCM* الگوریتم‌های خوشه‌یابی غیر خودکار هستند و تنها مراکز خوشه‌ها را ارائه می‌کنند. مقدار پارامتر F برای این مساله برابر است با ۳۱/۲۸۰۴ در حالی که مقدار بحرانی برای این مساله برابر است با ۱۳/۲۸. مقدار بحرانی مورد نظر از طریق جداول ارائه شده در [۳۵] به دست آمده است. بزرگ‌تر بودن مقدار F از مقدار بحرانی نشان از تفاوت عملکرد روش‌های مورد بررسی است و با توجه به جدول فوق با قاطعیت می‌توان رأی به برتری روش پیشنهادی داد.

۶- نتایج به دست آمده برای دو مجموعه کلان‌داده‌ی واقعی

دقت روش پیشنهادی در این پژوهش بر روی یک مجموعه کلان‌داده‌ی واقعی که شامل اطلاعات مکانی و حرکتی خودروهای سطح شهر پیزا می‌باشد، مورد بررسی قرار گرفته است. این داده‌ها با استفاده از گیرنده‌های GPS نصب شده بر روی خودروهای در حال حرکت جمع‌آوری شده‌اند. در حقیقت این مجموعه داده شامل سفرهای انجام شده توسط خودورها در یک مساحت و بازه‌ی زمانی مشخص است. هر کدام از این خودروها با یک شماره‌ی شناسایی منحصر به فرد مشخص شده‌اند. صاحبان این خودروها مشترکین یک شرکت بیمه هستند که مجاز به ذخیره‌سازی اطلاعات مکانی آن‌ها در هر لحظه جهت مبارزه با سرقت خودرو بوده است. این مجموعه داده که توسط شرکت Octo telematics جهت اهداف تحقیقاتی در اختیار محققین قرار گرفته است، شامل سفرهای طی شده توسط ۴۰۰۰۰ خودرو در طی ۵ هفته در شهر پیزا می‌باشد. موقعیت مکانی هر کدام از این خودروها هر ۳۰ ثانیه توسط دستگاه‌های GPS ضبط شده است. اطلاعات موجود در این مجموعه داده شامل شماره شناسایی خودرو، طول و عرض جغرافیایی و زمان می‌باشد.

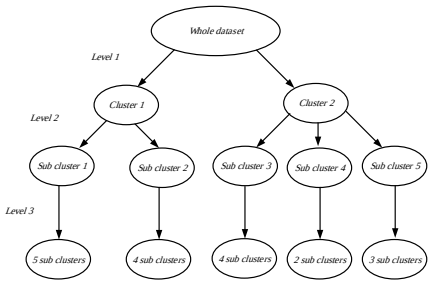
تمام سفرهای انجام شده توسط یک خودرو در زمان‌های مختلف در بازه‌های زمانی ۳۰ دقیقه‌ای در نظر گرفته شده‌اند. بدین صورت که اگر فاصله‌ی زمانی بین دو موقعیت مکانیِ متوالی و متمایز بیش از ۳۰ دقیقه باشد، موقعیت اول به عنوان نقطه‌ی اتمام سفر و موقعیت دوم به عنوان نقطه‌ی آغاز یک سفر دیگر در نظر گرفته شده است. با استفاده از این ساختار این مجموعه‌ی کلان‌داده شامل ۱۵۰۰۰۰۰ سفر می‌باشد که برای بررسی دقیق‌تر به هفت زیر مجموعه متناظر با روزهای هفته تقسیم‌بندی شده‌اند. در این پژوهش دو مجموعه کلان‌داده‌ی مربوط به سفرهای انجام شده در روزهای شنبه و یکشنبه بررسی شده‌اند. این دو مجموعه که با عناوین *Pisa_Monday* و *Pisa_Sunday* در این مقاله مورد بررسی قرار گرفته‌اند به ترتیب شامل ۴۹۰۰۰ و ۲۹۰۰۰ سفر می باشند که برای خوشه‌یابی تنها مختصات نقاط اول و آخر هر سفر در نظر گرفته شده است. به عبارت دیگر هر کدام به ترتیب شامل ۴۹۰۰۰ و ۲۹۰۰۰ داده در ۴ بعد می‌باشند که ابعاد آن‌ها طول و عرض جغرافیاییِ نقاط ابتدایی و انتهایی مسیرهای طی شده می‌باشند. برای بررسی دقیق‌تر و استخراج جزئیات بیش‌تر یک ساختار سلسله مراتبی از روش ارائه شده بر روی این دو مجموعه داده اعمال شده است بدین صورت که ابتدا در مرحله‌ی اول مجموعه‌داده‌ی مورد نظر توسط روش *AGWO-Clustering* خوشه‌بندی شده سپس در مرحله‌ی بعد هر کدام از این خوشه‌ها دوباره توسط روش ارائه شده مورد پردازش قرار گرفته و به تعدادی زیرخوشه تقسیم می‌شوند. این کار تا سه مرحله برروی خوشه‌های به دست آمده انجام شده و نتایج آن در جدول ۱۵ آورده شده است. با توجه به جدول ۱۵، روش *AGWO-Clustering* در مرحله‌ی اول کل مجموعه داده‌ی *Pisa_Monday* را در دو خوشه تقسیم بندی کرده است.

جدول ۱۵- نتایج به دست آمده برای مجموعه داده‌های *Pisa_Monday* و *Pisa_Sunday*

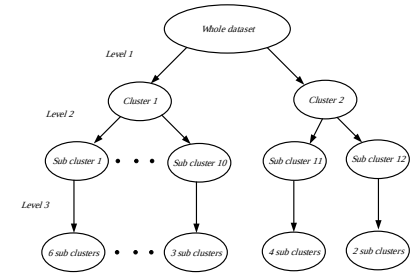
مجموعه داده	تعداد خوشه‌ها در مرحله‌ی اول	تعداد خوشه‌ها در مرحله‌ی دوم	تعداد خوشه‌ها در مرحله‌ی سوم
Pisa_Monday	۲	۵	۱۸
Pisa_Sunday	۲	۱۲	۶۲

در مرحله‌ی دوم این دو خوشه به پنج زیرخوشه‌ی دیگر تقسیم بندی شده و در مرحله‌ی آخر ۱۸ زیرخوشه از ۵ خوشه‌ی به دست آمده از مرحله‌ی دوم استخراج شده است. تعداد خوشه‌های متناظر با هر مرحله برای مجموعه داده‌ی *Pisa_Sunday* به ترتیب برابر با ۲، ۱۲ و ۶۲ می باشد. همان طور که مشخص است الگوریتم پیشنهادی تعداد خوشه‌های بیشتری برای مجموعه داده‌ی *Pisa_Sunday* پیدا کرده است که با توجه به تعطیل بودن روز یکشنبه علی رغم تعداد سفرهای کمتر انجام شده در این روز، این مساله منطقی به نظر می‌رسد. در تصویر شماره‌ی ۱۵

فرایند خوشه‌بندی این دو مجموعه داده نشان داده شده است. همچنین جزئیات بیشترِ نتایج استخراج شده در دو زیر بخش بعدی آورده شده‌اند.



الف

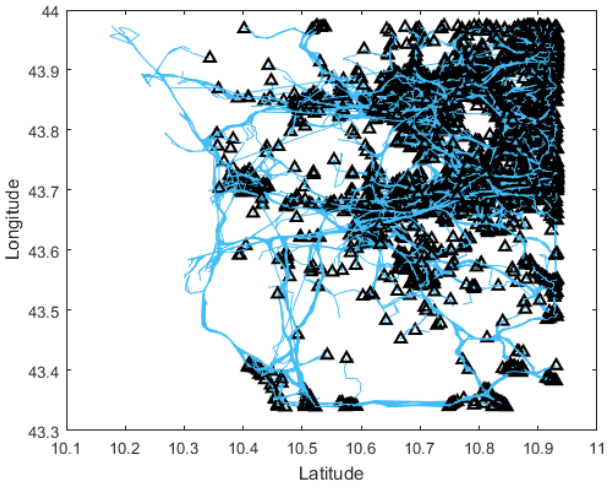


ب

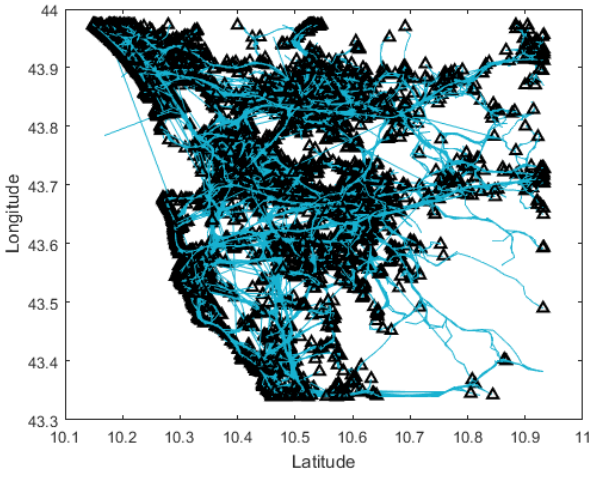
شکل ۱۵ الف و ب- فرایند خوشه‌یابی مجموعه داده‌های Pisa_Monday و Pisa_Sunday

۱-۶- نتایج به دست آمده برای مجموعه کلان‌داده‌ی *Pisa_Monday*

در تصویر شماره‌ی ۱۶ دو خوشه‌ی اصلی یافت شده برای این مجموعه داده نشان داده شده است. در این تصاویر خطوط رسم شده مشخص کننده‌ی مسیر طی شده هستند و مثلث‌های سیاه رنگ انتهای مسیر را نشان می‌دهند. با توجه به این تصویر، به طور کلی خوشه‌ی اول (شکل ۱۶ الف) دربردارنده‌ی سفرهای انجام شده به سمت شرق است در حالی که خوشه‌ی دوم (شکل ۱۶ ب) دربردارنده‌ی سفرهای انجام شده به سمت غرب می‌باشد. در تصاویر ۱۷ و ۱۸ نیز سه نمونه از زیرخوشه‌های یافت شده در مرحله‌ی ۳ برای هر کدام از این دو خوشه‌ی اصلی نشان داده شده‌اند. همان طور که مشخص است زیر خوشه‌های نشان داده شده در تصاویر ۱۷ و ۱۸ نیز به طور کلی شامل سفرهایی به ترتیب به مقصد شرق و غرب می‌باشند. به زبان ساده‌تر الگوریتم پیشنهادی سفرهای با مقصد یکسان را در یک خوشه قرار داده است.

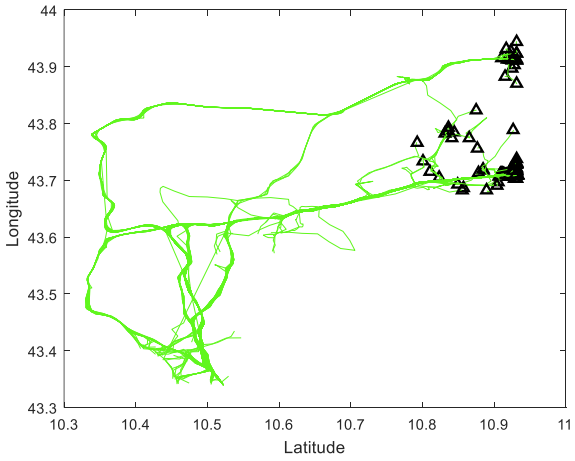


الف

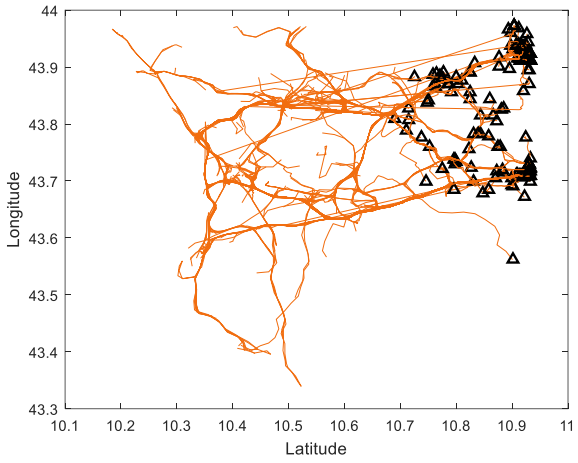


ب

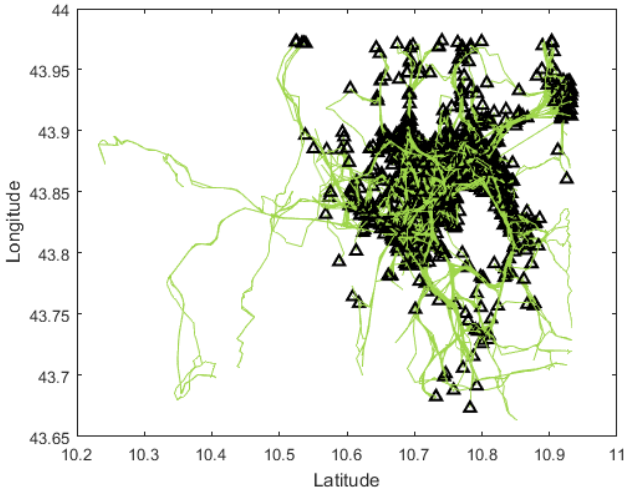
شکل ۱۶ الف و ب- دو خوشه‌ی اصلی یافت شده برای مجموعه داده‌ی Pisa_Monday



الف

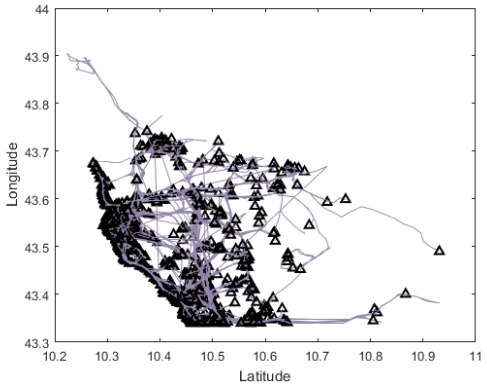


ب

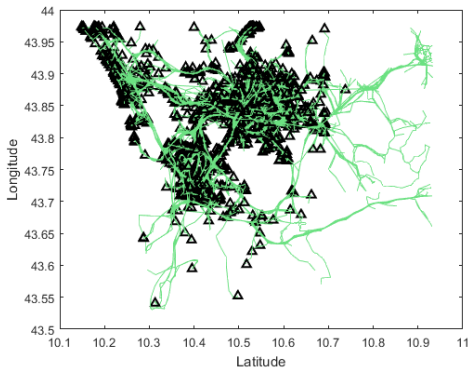


ج

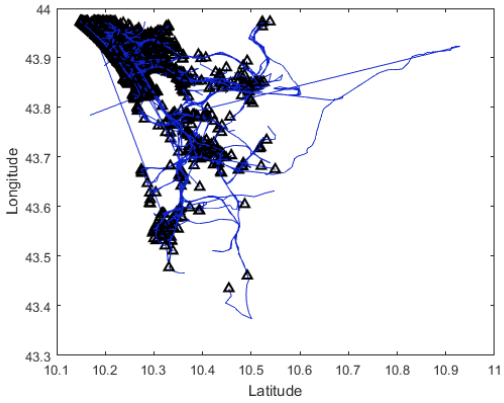
شکل ۱۷ الف، ب و ج- سه زیرخوشه‌ی متعلق به خوشه‌ی اصلی اول برای مجموعه داده‌ی Pisa_Monday



الف



ب



ج

شکل ۱۸ الف، ب و ج- سه زیرخوشه‌ی متعلق به خوشه‌ی اصلی دوم برای مجموعه داده‌ی Pisa_Monday

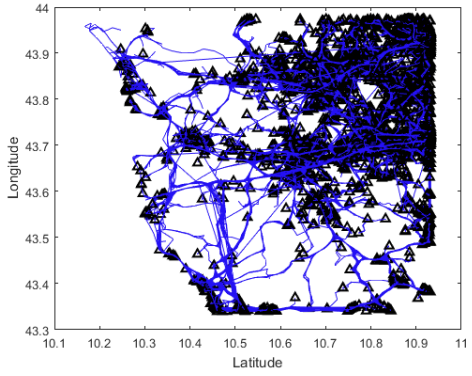
۲-۶- نتایج به دست آمده برای مجموعه کلان‌داده‌ی *Pisa_Sunday*

همانند مجموعه داده‌ی *Pisa_Monday*، این مجموعه داده نیز توسط روش پیشنهادی در مرحله‌ی اول به دو خوشه تقسیم شده است. در مراحل بعدی نیز به ترتیب ۱۲ و ۶۲ خوشه استخراج شده‌اند. در تصویر ۱۹ دو خوشه‌ی اصلی یافت شده در مرحله‌ی اول نشان داده شده‌اند. به طور کلی برای این مجموعه داده نیز خوشه‌ی اول شامل تمام مسیرهای پیموده شده به سمت شرق است در حالی که خوشه‌ی دوم دربردارنده ی مسیرهای منتهی به غرب می‌باشد. در تصاویر ۲۰ و ۲۱ نیز سه نمونه از زیرخوشه‌های استخراج شده از هر کدام از این دو خوشه‌ی اصلی نشان داده شده‌اند. آنچه از جدول ۱۵ و تصاویر ۱۶ تا ۲۱ استنتاج می‌شود این است که روش پیشنهادی، سفرهای با مقصد یکسان را با دقت بالایی پیدا کرده و آن‌ها را در یک خوشه قرار می‌دهد. به طور کلی عملکرد روش ارائه شده در این پژوهش، در خوشه‌یابی کلان‌داده‌ها و همچنین دقت آن در پیدا کردن تعداد خوشه‌ها بسیار قابل توجه می‌باشد.

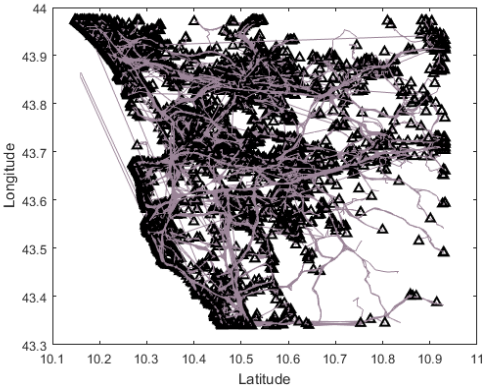
۷- نتیجه گیری

خوشه‌یابی یکی از روش‌های مهم داده‌کاوی و همچنین یکی از روش‌های بسیار با اهمیت در فرایند استخراج اطلاعات از کلان‌داده‌ها می‌باشد.

روش‌های مختلفی تاکنون برای خوشه‌یابی ارائه شده‌اند. از جمله روش *Kmeans* که هم اکنون نیز جزء پر استفاده‌ترین روش‌های خوشه‌یابی است. نقاط ضعف روش *Kmeans* عبارتند از: احتمال بالای گیر افتادن در نقطه‌ی بهینه‌ی محلی، وابستگی پاسخ نهایی به پاسخ‌های اولیه که به صورت تصادفی ایجاد می‌شوند و مهم‌ترین آن‌ها، عدم توانایی در پیدا کردن تعداد خوشه‌ها. این نقاط ضعف مانع از عملکرد خوب این روش در مواجهه با مجموعه داده‌های بزرگ می‌شود به خصوص این که یافتن تعداد خوشه‌هایِ یک مجموعه داده از اهمیت بالایی برخوردار است. در این پژوهش یک روش خوشه‌یابی خودکار با استفاده از الگوریتم ابتکاری گرگ خاکستری ارائه شده است که دقت بالایی در پیدا کردن تعداد خوشه‌ها دارد.

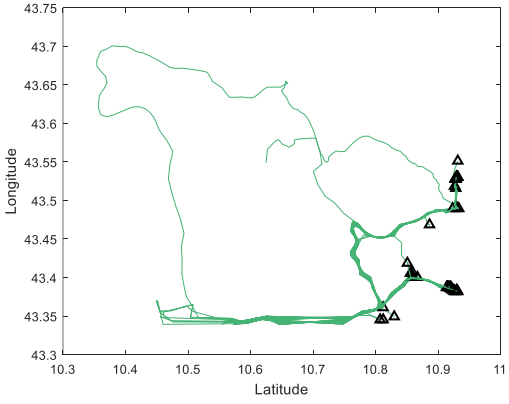


الف

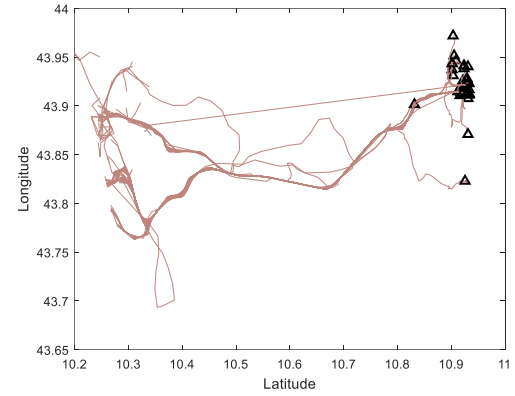


ب

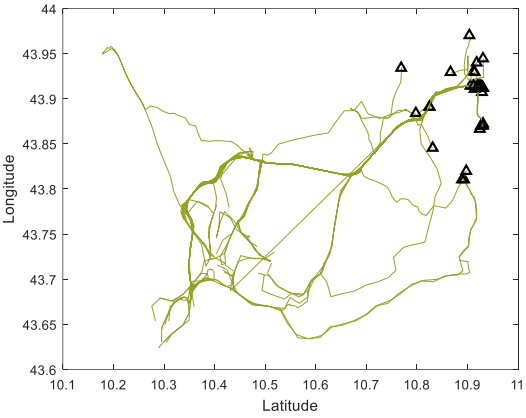
شکل ۱۹ الف و ب- دو خوشه‌ی اصلی یافت شده برای مجموعه داده‌ی Pisa_Sunday



الف



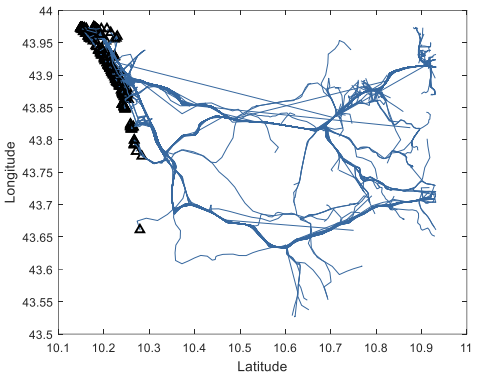
ب



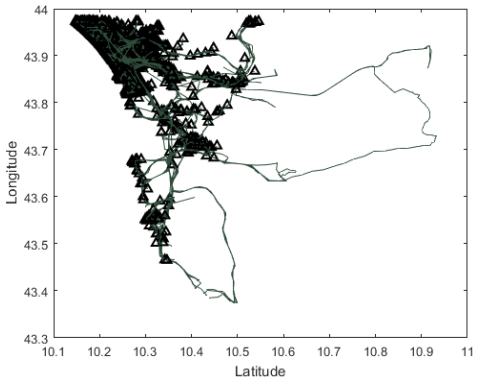
ج

شکل ۲۰ الف، ب و ج- سه زیرخوشه‌ی متعلق به خوشه‌ی اصلی اول برای مجموعه داده‌ی Pisa_Sunday

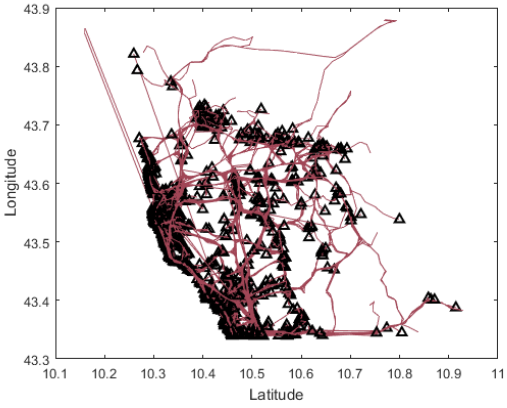
عملکرد روش پیشنهادی در دو حالت خوشه‌یابی غیر خودکار و خوشه‌یابی خودکار بر روی ۱۳ مجموعه داده‌ی مصنوعی و دو مجموعه ی کلان‌داده‌ی واقعی مورد ارزیابی قرار گرفته و با روش‌های *Kmeans* و *Xmeans* نیز مقایسه شده است. نتایج به دست آمده برای داده‌های مصنوعی که در جداول ۲ تا ۱۳ و همچنین شکل‌های ۶ تا ۱۴ آورده شده‌اند نشان از توان بالای روش پیشنهادی در خوشه‌یابی کلان‌داده‌ها و همچنین برتری آن بر روش‌های *Kmeans* و *Xmeans* می‌باشد. همچنین در این جداول عملکرد الگوریتم گرگ خاکستری با الگوریتم ازدحام ذرات که یکی از محبوب‌ترین الگوریتم‌های ابتکاری نیز می‌باشد، مقایسه گردیده است. اعداد درج شده در جداول و همچنین نمودارهای همگرایی نشان از برتری الگوریتم گرگ خاکستری بر الگوریتم ازدحام ذرات در این مساله‌ی خاص دارد. با این وجود نمی‌توان با قاطعیت مدعی شد که این روش ابتکاری در تمامی مسائل بهینه‌سازی برتر از الگوریتم ازدحام ذرات می‌باشد. همان طور که در مورد هیچ الگوریتم ابتکاری دیگری نیز نمی‌توان چنین ادعایی کرد. در حقیقت جداول و تصاویر بخش‌های قبل نشان از توان بالای الگوریتم‌های ابتکاری در پردازش کلان‌داده‌ها به خصوص در خوشه‌یابی خودکار این نوع داده‌ها دارند.



الف



ب



ج

شکل ۲۱ الف، ب و ج- سه زیرخوشه‌ی متعلق به خوشه‌ی اصلی دوم برای مجموعه داده‌ی Pisa_Sunday

علاوه بر این، نتایج حاصل شده از اعمال این روش بر روی دو مجموعه کلان‌داده‌ی واقعی که در جدول ۱۵ و تصاویر ۱۶ تا ۲۱ آورده شده‌اند توان بالای این روش را در خوشه‌یابی کلان‌داده‌ها و همچنین یافتن تعداد خوشه‌ها نشان می‌دهند.

۸- کارهای آینده

در آینده می‌توان برای ادامه‌ی این راه از روش‌های موازی سازی جهت اجرای موازی روش ارائه شده در هر دو فاز یعنی جست‌وجو برای یافتن تعداد خوشه‌ها و همچنین جست‌وجو برای یافتن موقعیت مراکز خوشه‌ها، استفاده کرد. همچنین با توجه به این که طول عوامل جست‌وجو متناسب با تعداد بردارهای ویژگی مجموعه داده‌ی مورد نظر می‌باشد و همچنین در هر بار محاسبه‌ی تابع برازندگی تمام داده‌ها باید برچسب زده شوند، جهت کم کردن حجم محاسبات می‌توان از روش‌های ابتکاری برای کم کردن تعداد نمونه‌ها و همچنین تعداد ابعاد به صورت هوشمند استفاده کرد.

مراجع

[1] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, "Big data, analytics and the path from insights to value," MIT sloan management review, vol. 52, p. 21, 2011.

[2] S. Cheng, Y. Shi, Q. Qin, and R. Bai, "Swarm intelligence in big data analytics," in International Conference on Intelligent Data Engineering and Automated Learning, 2013, pp. 417-426.

[3] J. Leskovec, A. Rajaraman, and J. D. Ullman, Mining of Massive Datasets: Cambridge University Press, 2014.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," ACM computing surveys (CSUR), vol. 31, pp. 264-323, 1999.

[5] J. A. Hartigan, "Clustering algorithms (probability & mathematical statistics)," ed: John Wiley & Sons Inc New York, 1975.

[6] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Transactions on neural networks, vol. 16, pp. 645-678, 2005.

[7] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, pp. 100-108, 1979.

[۸] احسان نادرنژاد، حمید حسن‌پور و حسین میارنعیمی، «استفاده از مشخصه‌های آماری داده‌ها و پردازش بلوکی برای قطعه بندی تصاویر»، فصلنامه مهندسی برق دانشگاه تبریز، دوره‌ی ۳۹ شماره‌ی ۱، صفحه‌ی ۴۸ تا ۵۷، بهار ۱۳۸۸.

[9] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: a review," in International Conference on Computational Science and Its Applications, 2014, pp. 707-720.

[10] R. C. Eberhart, Y. Shi, and J. Kennedy, Swarm intelligence: Elsevier, 2001.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46-61, 2014.

[12] A. Sinha and P. K. Jana, "A novel K-means based clustering algorithm for big data," in Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on, 2016, pp. 1875-1879.

[13] M. Jain and C. Verma, "Adapting k-means for Clustering in Big Data," International Journal of Computer Applications, vol. 101, pp. 19-24, 2014.

[14] A. Saini, J. Minocha, J. Ubriani, and D. Sharma, "New approach for clustering of big data: DisK-means," in Computing, Communication and Automation (ICCCA), 2016 International Conference on, 2016, pp. 122-126.

[15] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," presented at the Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, Louisiana, 2007.

[16] S. H. Razavi, E. O. M. Ebadati, S. Asadi, and H. Kaur, "An efficient grouping genetic algorithm for data clustering and big data analysis," in Computational Intelligence for Big Data Analysis, ed: Springer, 2015, pp. 119-142.

[17] S. Saitta, B. Raphael, and I. F. Smith, "A comprehensive validity index for clustering," Intelligent Data Analysis, vol. 12, pp. 529-548, 2008.

[18] A. Abraham, S. Das, and S. Roy, "Swarm intelligence algorithms for data clustering," in Soft computing for knowledge discovery and data mining, ed: Springer, 2008, pp. 279-313.

[19] S. H. Kwon, "Cluster validity index for fuzzy clustering," Electronics letters, vol. 34, pp. 2176-2177, 1998.

[20] X. Cui, T. E. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, 2005, pp. 185-191.

[21] M. G. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," Pattern Analysis and Applications, vol. 8, p. 332, 2006.

[22] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 1650-1654, 2002.

[23] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," Expert Systems with Applications, vol. 37, pp. 4761-4767, 2010.

[24] G. Krishnasamy, A. J. Kulkarni, and R. Paramesran, "A hybrid approach for data clustering based on modified cohort intelligence and K-means," Expert Systems with Applications, vol. 41, pp. 6009-6016, 2014.

[25] Y. Lu, B. Cao, C. Rego, and F. Glover, "A Tabu search based clustering algorithm and its parallel implementation on Spark," Applied Soft Computing, vol. 63, pp. 97-109, 2018.

[26] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," Applied soft computing, vol. 11, pp. 652-657, 2011.

[27] X. Han, L. Quan, X. Xiong, M. Almeter, J. Xiang, and Y. Lan, "A novel data clustering algorithm based on modified gravitational search algorithm," Engineering Applications of Artificial Intelligence, vol. 61, pp. 1-7, 2017.

[28] A. Banharnsakun, "A MapReduce-based artificial bee colony for large-scale data clustering," Pattern Recognition Letters, vol. 93, pp. 78-84, 2017.

[29] C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations," Behavioural processes, vol. 88, pp. 192-197, 2011.

[30] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," Communications in Statistics-theory and Methods, vol. 3, pp. 1-27, 1974.

[31] School of Computing University of Eastern Finland. "clustering basic benchmarks," June 10, 2018; https://cs.joensuu.fi/sipu/datasets/.

[32] W. M. Rand, "Objective criteria for the evaluation of clustering methods," Journal of the American Statistical association, vol. 66, pp. 846-850, 1971.

[33] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in Icml, 2000, pp. 727-734.

[34] Z. F. Knops, J. A. Maintz, M. A. Viergever, and J. P. Pluim, "Normalized mutual information based registration using k-means clustering and shading correction," Medical image analysis, vol. 10, pp. 432-439, 2006.

[35] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," Journal of Machine learning research, vol. 7, pp. 1-30, 2006.

[36] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," Pattern Recognition, vol. 39, pp. 761-775, 2006.

[37] I. Kärkkäinen and P. Fränti, Dynamic local search algorithm for the clustering problem: University of Joensuu, 2002.

[38] P. Fränti, R. Mariescu-Istodor, and C. Zhong, "XNN graph," in Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), 2016, pp. 207-217.

[39] P. Franti, O. Virmajoki, and V. Hautamaki, "Fast agglomerative clustering using a k-nearest neighbor graph," IEEE transactions on pattern analysis and machine intelligence, vol. 28, pp. 1875-1881, 2006.

[۴۰] سمیرا رفیعی، پرهام مرادی، «بهبود عملکرد الگوریتم خوشه‌بندی فازی سی-مینز با وزن‌دهی اتوماتیک و محلی ویژگی‌ها»، مجله‌ی مهندسی برق دانشگاه تبریز، دوره‌ی ۴۶، صفحه‌ی ۷۵ تا ۸۶، تابستان ۱۳۹۵.

[41] I. Aljarah and S. A. Ludwig, "Parallel particle swarm optimization clustering algorithm based on mapreduce methodology," in Nature and biologically inspired computing (NaBIC), 2012 fourth world congress on, 2012, pp. 104-111.

[42] B. Wu, G. Wu, and M. Yang, "A mapreduce based ant colony optimization approach to combinatorial optimization problems," in Natural Computation (ICNC), 2012 Eighth International Conference on, 2012, pp. 728-732.

[43] J. Li, X. Hu, Z. Pang, and K. Qian, "A parallel ant colony optimization algorithm based on fine-grained model with GPU-acceleration," International Journal of Innovative Computing, Information and Control, vol. 5, pp. 3707-3716, 2009.

[44] D.-W. Huang and J. Lin, "Scaling populations of a genetic algorithm for job shop scheduling problems using MapReduce," in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, 2010, pp. 780-785.

زیرنویس‌ها

[1] Big data analytics
[2] Large scale optimization
[3] Artificial bee colony
[4] Cohort Intelligence
[5] Meta heuristic
[6] Exploration
[7] Exploitation
[8] Overall between cluster variance
[9] Overall within cluster variance
[10] Normalized Mutual Information
[11] Friedman test