خوشه‌بندی داده با استفاده از ترکیب K-harmonic means و PSO

ملیحه دانش[۱]، محمدحسین یغمایی مقدم[۲] و محمدرضا اکبرزاده توتونچی[۳]

[۱]باشگاه پژوهشگران جوان دانشگاه آزاد اسلامی واحد جویبار، ma.danesh@stu-mail.um.ac.ir

[۲] دانشگاه فردوسی مشهد ، yaghmaee@ieee.org

[۳] دانشگاه فردوسی مشهد، akbarzadeh@ieee.org

چکیده – خوشه‌بندی داده یکی از رایج‌ترین تکنیک‌های داده‌کاوی است. الگوریتم *K-means* یکی از مشهورترین الگوریتم‌های خوشه‌بندی داده می‌باشد که به دلیل پیاده‌سازی آسان و سرعت عملکرد، محبوبیت زیادی یافته است اما مشکلاتی از قبیل حساس بودن به مقدار اولیه و گرفتار شدن در دام بهینه محلی از قدرت عملکرد آن می‌کاهد. روش خوشه‌بندی *K-harmonic means*، مسئله حساس بودن به مقدار اولیه را پوشش می‌دهد اما مشکل گرفتار شدن در دام بهینه محلی همچنان این الگوریتم را تهدید می‌کند. الگوریتم *Particle Swarm Optimization* یک تکنیک بهینه‌سازی سراسری احتمالی است که راه‌حل مناسبی برای غلبه بر مشکل ذکر شده می‌باشد. در این مقاله الگوریتم ترکیبی *PSOKHM* سعی می‌کند که با بهره‌گیری از مزایای هر دو الگوریتم، علاوه بر فرار از گرفتار شدن در دام بهینه محلی در الگوریتم *KHM*، بر سرعت پایین همگرایی الگوریتم *PSO* نیز غلبه کند. روش پیشنهادی در این مقاله، ترکیب *PSO* باالگوریتم تکاملی ژنتیک در *PSOKHM* می‌باشد، *GSOKHM*، که در جهت کارکرد مناسب تر الگوریتم *PSO* پیشنهاد شده‌است. برای انجام آزمایش از چهار مجموعه داده واقعی استفاده شده است که نتایج به دست آمده حاکی از عملکرد بهتر این روش در امر خوشه‌بندی نسبت به الگوریتم ترکیبی *PSOKHM* می‌باشد.

کلید واژه- خوشه‌بندی داده، KHM، PSO، الگوریتم ژنتیک.

۱- مقدمه

یکی از روش‌های حیاتی کنترل و مدیریت داده‌ها، خوشه بندی داده‌های با خواص مشابه، درون مجموعه‌ای از دسته‌ها می باشد. خوشه‌بندی فرآیندی است که در آن مجموعه‌ای از اشیاء داده به گروه‌های مجزایی از کلاس‌ها، خوشه، تقسیم می‌شوند به طوری‌که اشیای یک کلاس تا حد امکان به یکدیگر شبیه بوده و با اشیاء دیگر کلاس‌ها، متفاوت می‌باشند. خوشه‌بندی در زمینه های بسیاری از جمله در شناسایی الگو، یادگیری ماشین، داده کاوی، بازیابی اطلاعات و انفورماتیک زیستی کاربرد دارد. یکی از کاربردی‌ترین روش‌های خوشه‌بندی، روش خوشه‌بندی K-means (KM) است. هدف اصلی خوشه‌بندی KM این است که مجموع عدم تشابه بین تمام اشیاء یک خوشه از مراکز خوشه‌های متناظرشان کمترین باشد. مهم‌ترین مشکل الگوریتم KM این است که نتایج خوشه‌بندی حساس به انتخاب مراکز خوشه‌های اولیه می‌باشند و ممکن است به بهینه محلی همگرا شوند [۱و۵]. الگوریتم k-harmonic means (KHM) الگوریتمی است که در سال ۲۰۰۲ پیشنهاد شد و هدف از آن این بود که میانگین هارمونیک تمامی نقاط موجود در یک مجموعه داده را تا مراکز خوشه‌ها کمترین کند. الگوریتم KHM به حل مشکل مقداردهی اولیه در الگوریتم KM می‌پردازد اما همچنان مشکل گرفتار شدن در دام بهینه محلی آن را تهدید می‌کند. بنابراین به هدف دستیابی به الگوریتم خوشه‌بندی بهتر، باید به دنبال راه‌حلی برای غلبه بر مشکل افتادن در دام بهینه محلی باشیم[۱و۳].

بهینه‌سازی گروهی ذرات (PSO)، یک تکنیک بهینه‌سازی مبتنی بر جمعیت می‌باشد که از رفتار گروهی پرندگان و ماهی‌ها الهام گرفته شده است. از این الگوریتم می‌توان به عنوان روشی برای کمک به الگوریتم KHM جهت فرار از گرفتار شدن در دام بهینه محلی استفاده کرد که الگوریتم خوشه‌بندی ترکیبی PSOKHM سعی می‌کند از مزایای هر دو روش جهت بهبود عمل خوشه‌بندی بهره ببرد. روش پیشنهادی ما حاصل از ترکیب PSO با الگوریتم تکاملی ژنتیک در الگوریتم PSOKHM جهت بهبود عملکرد PSO می‌باشد. همچنین به منظور بررسی کارایی الگوریتم پیشنهادی، از چهار مجموعه داده واقعی استفاده شده

است [۲]. در ادامه‌ی مباحث، در بخش دوم به تشریح الگوریتم ترکیبی PSOKHM پرداخته و در آن هر یک از الگوریتم‌های PSO و KHM را به طور خلاصه توضیح می‌دهیم. در بخش سوم الگوریتم پیشنهادی GSOKHM را معرفی می‌کنیم و در بخش چهارم نتایج حاصل از اعمال روش پیشنهادی را بر روی چهار مجموعه داده واقعی آورده و به مقایسه‌ی آن با روش‌های پیشین می‌پردازیم. نهایتا در قسمت پنجم خلاصه‌ای از کار انجام شده در این تحقیق را توضیح می‌دهیم.

۲- الگوریتم خوشه‌بندی ترکیبی PSOKHM

برای توضیح الگوریتم ترکیبی فوق ابتدا هر یک از الگوریتم های PSO و KHM را به طور خلاصه شرح می‌دهیم وسپس به توصیف الگوریتم PSOKHM می‌پردازیم.

۲-۱- الگوریتم K-harmonic means

خوشه‌بندی KM یک روش ساده و سریع است که به دلیل پیاده‌سازی آسان و تعداد تکرار کم، عموما مورد استفاده قرار می گیرد. الگوریتم KM در تلاش برای یافتن مراکز خوشه‌های $(c_1,c_2,...,c_k)$ به گونه‌ای عمل می‌کند که مجموع مربعات فاصله‌ی هر نقطه x_i تا نزدیک‌ترین مرکز خوشه(c_j) کمترین شود. وابستگی کارایی KM روی مقداردهی اولیه مراکز، یک مشکل اصلی این الگوریتم می‌باشد. در این الگوریتم ارتباط قوی‌ای بین نقاط داده و نزدیک‌ترین مراکز خوشه برقرار شده و باعث می‌شود مراکز خوشه‌ها از محدوده‌ی تراکم محلی داده‌ها خارج نشوند. روش KHM این مشکل عمده را از طریق جایگزینی کمترین فاصله یک نقطه از مراکز که در KM استفاده می‌شود با میانگین هارمونیک فاصله هر نقطه تا تمامی مراکز برطرف می‌کند. میانگین هارمونیک یک امتیاز مناسبی را به هر نقطه‌ی داده بر اساس نزدیکی آن به هر مرکز می‌دهد که این امر را به عنوان یک ویژگی میانگین هارمونیک در نظر می‌گیرند.

نمادهای زیر برای فرمول‌بندی الگوریتم KHM استفاده می‌شود:

داده‌ای که باید خوشه‌بندی شود. $X = \{x_1, x_2, ..., x_n\} =$

مجموعه مراکز خوشه‌ها $C = \{c_1, c_2, ..., c_k\} =$

تابع عضویتی که سهم داده‌ی x_i را که متعلق به مرکز c_j است، تعریف می‌کند. $m(c_j \mid x_i) =$

تابع وزنی که میزان تاثیر داده x_i را در محاسبه‌ی مجدد پارامترهای مرکز در تکرار بعدی تعریف می‌کند. $w(x_i) =$

الگوریتم پایه برای خوشه‌بندی KHM به صورت زیر می‌باشد:

۱- مقداردهی اولیه الگوریتم با مراکز حدسی C (انتخاب تصادفی مراکز).

۲- محاسبه مقدار تابع هدف به صورت زیر است:

$$KHM(X,C) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{K} \frac{1}{\| x_i - c_j \|^p}} \quad (۱)$$

که p یک پارامتر ورودی با مقدار $p \geq 2$ می‌باشد.

۳- برای هر داده x_i، تابع عضویت $m(c_j|x_i)$ به ازای هر مرکز c_j به صورت زیر محاسبه می‌شود:

$$m(c_j \mid x_i) = \frac{\| x_i - c_j \|^{-p-2}}{\sum_{j=1}^{k} \| x_i - c_j \|^{-p-2}} \quad (۲)$$

۴- برای هر داده x_i، وزن $w(x_i)$ مربوط به آن به صورت زیر محاسبه می‌شود:

$$w(x_i) = \frac{\sum_{j=1}^{k} \| x_i - c_j \|^{-p-2}}{\left(\sum_{j=1}^{k} \| x_i - c_j \|^{-p} \right)^2} \quad (۳)$$

۵- برای هر مرکز c_j، فاصله آن از تمامی نقاط x_i بر طبق توابع عضویت و وزن‌هایشان به صورت زیر محاسبه مجدد می‌شود:

$$c_j = \frac{\sum_{i=1}^{n} m(c_j \mid x_i) w(x_i) x_i}{\sum_{i=1}^{n} m(c_j \mid x_i) w(x_i)} \quad (۴)$$

۶- گام‌های ۲ تا ۵ را به ازای تعداد تکرار از پیش تعریف شده‌ای انجام می‌دهیم یا تا زمانی که KHM(X,C) به اندازه قابل توجه ای تغییر نکند.

۷- نقطه x_i را به کلاستر j با بزرگترین $m(c_j|x_i)$ تخصیص می دهیم.

این الگوریتم نشان می‌دهد که KHM ضرورتا به مقداردهی اولیه مراکز حساس نمی‌باشد ولی تمایل به همگرا شدن به بهینه محلی در آن وجود دارد[۱و۳و۱۲].

۲-۲- بهینه سازی گروهی ذرات (PSO)

روش PSO توسط کندی و ابرهارت در سال ۱۹۹۵ توسعه یافت و تاکنون به طور موفقیت آمیزی در خیلی از زمینه‌های علوم و کاربردی مورد استفاده قرار گرفته است. PSO یک

الگوریتم بهینه‌سازی مبتنی بر جمعیت می‌باشد که در آن هر فرد به عنوان یک ذره در نظر گرفته می‌شود و هر جمعیت از تعدادی از این ذرات تشکیل شده است. در PSO فضای حل مسئله به عنوان فضای جستجو در نظر گرفته می‌شود و هر مکان در فضای جستجو یک راه‌حل وابسته به مسئله می‌باشد. در این جمعیت، ذرات سعی می‌کنند که با همکاری یکدیگر بهترین موقعیت (بهترین راه‌حل) را در فضای جستجو (فضای راه‌حل) پیدا کنند. همچنین هر ذره مطابق با سرعتش حرکت می‌کند. حرکت هر ذره در هر تکرار با فرمول زیر محاسبه می‌شود:

$$x_i(t+1) \leftarrow x_i(t) + v_i(t) \quad (۵)$$

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 rand_1(pbest_i(t) - x_i(t)) + c_2 rand_2(gbest(t) - x_i(t)) \quad (۶)$$

در معادلات (۵) و (۶)، $x_i(t)$ موقعیت ذره‌ی iام در زمان t می باشد و $v_i(t)$ سرعت ذره‌ی iام در زمان t می‌باشد. $pbest_i(t)$ بهترین موقعیتی است که توسط خود ذره iام تاکنون پیدا شده است. gbest(t) بهترین موقعیتی است که تاکنون توسط کل جمعیت پیدا شده است. ω یک وزن اینرسی است که نسبتی از سرعت پیشین را می‌دهد و c_1 و c_2 نیز که ضرایب شتاب می باشند، تاثیر بهترین موقعیت هر ذره و بهترین موقعیت سراسری را تعیین می‌کنند. همچنین $rand_1$ و $rand_2$ متغیرهای تصادفی بین ۰ و ۱ هستند. روال الگوریتم PSO در شکل (۱) آمده است [۵و۸].

```
Initialize a population of particles with random positions and
velocities in the search space.
While(termination conditions are not met)
{
   For each particle i do
      Update the position of particle i according t equation
      (5).
      Update the velocity of particle i according to equation
      (6).
      Map the position of particle i in the solution space
      and evaluate its fitness value according to the fitness
      function.
      Update pbesti(t) and gbest ( t) if necessary.
   End for
}
```

شکل ۱: شبه‌کد الگوریتم PSO

۲-۳- الگوریتم PSOKHM

الگوریتم KHM سرعت همگرایی بیشتری از الگوریتم PSO دارد چرا که به ارزیابی تابعی کمتری نیاز دارد اما معمولا به دلیل طبیعت جستجوی حریصانه‌اش در دام بهینه محلی گرفتار می شود. الگوریتم خوشه‌بندی ترکیبی PSOKHM سعی می‌کند با ترکیب KHM و PSO با هم، از مزایای هر دوی آنها بهره‌مند گردد. این الگوریتم ترکیبی، KHM را به تعداد چهار بار در هر نسل تکرار می‌کند و بدین منظور از ۸ نسل برای بهبود ذرات در جمعیت استفاده می‌کند. همچنین الگوریتم PSO نیز در هر نسل به تعداد ۸ بار تکرار می‌شود.

یک ذره برداری از اعداد واقعی با ابعاد k*d است به طوریکه k تعداد خوشه‌ها و d ابعاد داده‌ای است که باید خوشه‌بندی گردد. در شکل(۲) نمونه‌ای از یک ذره در جمعیت آورده شده است. تابع ارزیابی آن، تابع هدف الگوریتم KHM می‌باشد. در شکل(۳) خلاصه‌ای از الگوریتم PSOKHM نشان داده شده است. همان طور که از شکل پیداست، در هر نسل ابتدا الگوریتم PSO به تعداد تکرارهای گفته شده بر روی ذرات اعمال می‌شود و به دنبال آن الگوریتم KHM بر روی نتایج حاصل از PSO تکرار می‌گردد[۱] .

x_{11}	x_{12}	...	x_{1d}	...	X_{k1}	X_{k2}	...	X_{kd}

شکل ۲: نمایش یک ذره

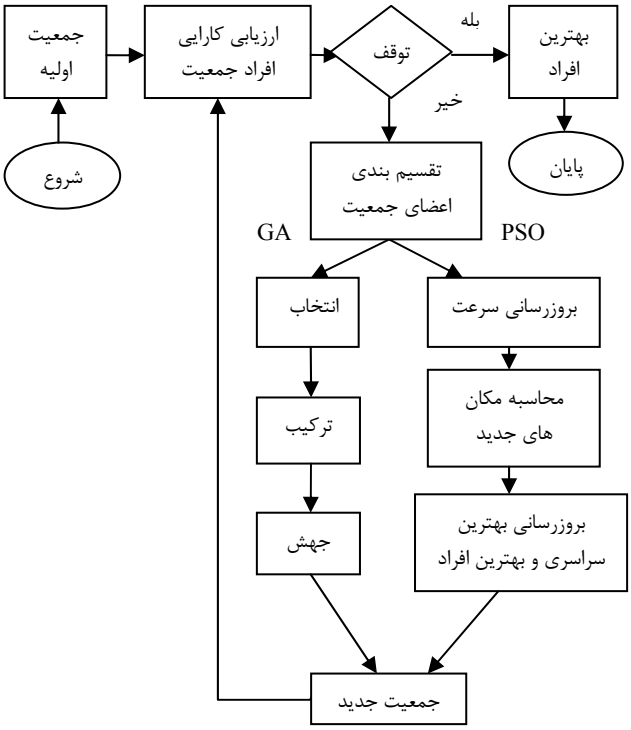
```
Step 1: Set the initial parameters including the maximum
iterative count IterCount, the population size Psize, ω, c1 and
c2.
Step 2: Initialize a population of size Psize.
Step 3: Set iterative count Gen1= 0.
Step 4: Set iterative count Gen2= Gen3=0.
Step 5 (PSO Method)
Step 5.1: Apply the PSO operator to update the Psize
particles.
Step 5.2: Gen2=Gen2+1. If Gen2<8, go to Step 5.1.
Step 6 (KHM Method) For each particle i do
Step 6.1: Take the position of particle i centers of the KHM
algorithm.
Step 6.2 : Recalculate each cluster center using the KHM
algorithm.
Step6.3: Gen3=Gen3+1. If Gen3<4, go to Step 6.2.
Step 7: Gen1= Gen1+1. If Gen1<IterCount, go to Step 4.
Step 8: Assign data point i x to cluster j with the biggest
m(cj|xi).
```

شکل ۳: الگوریتم خوشه‌بندی ترکیبی PSOKHM

۳- روش پیشنهادی GSOKHM

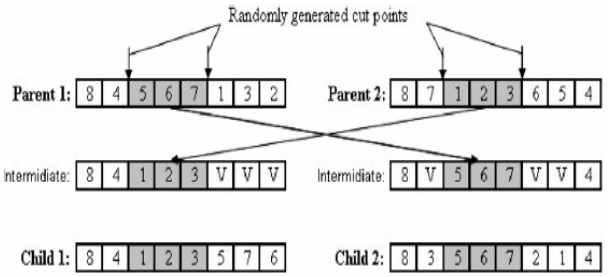
در این روش سعی کردیم که به منظور افزایش کارایی الگوریتم PSO در PSOKHM آن را با یک روند تکاملی دیگر

نظیر الگوریتم ژنتیک ترکیب کنیم و به این ترتیب در جهت خوشه‌بندی هر چه بهتر داده‌ها قدم برداریم. الگوریتم ژنتیک یکی از انواع الگوریتم‌های تصادفی است که از عملگرهای انتخاب، تقاطع و جهش استفاده می‌نماید. این الگوریتم از مشهورترین الگوریتم‌های تکاملی می‌باشد و به صورت گسترده‌ای در حل مسائل بهینه‌سازی مشکل مورد استفاده قرار گرفته است. الگوریتم ژنتیک به منظور رفع بهینه محلی در الگوریتم KHM و کمک به افزایش کارایی الگوریتم PSO می‌تواند بسیار موثر باشد. جهت استفاده از ترکیب الگوریتم‌های PSO و GA در کاربرد مورد نظر، از الگوریتم GSO بطوریکه عملکرد آن در نمودار شکل (۴) آمده‌است، استفاده نموده‌ایم.



شکل۴: نمودار الگوریتم ترکیبی GSO

همان‌طور که از نمودار بالا پیداست، در هر تکرار اعضای جمعیت به دو قسمت با اندازه‌ی مساوی تقسیم شده و عملگرهای PSO و GA به طور مستقل روی هر قسمت اعمال می‌شود و نهایتا برای ارزیابی تغییرات، مجددا با هم ترکیب می شوند. این روال تا رسیدن به شرایط مطلوب ادامه دارد. همچنین از روش انتخاب چرخ رولت در انجام عمل selection در الگوریتم ژنتیک بهره بردیم و عمل ترکیب (crossover) نیز مطابق با شکل (۵) انجام می‌گیرد. به منظور انجام عمل جهش (mutation) نیز نقاطی از ذرات تصادفی در هر نسل را به صورت احتمالی انتخاب می‌کنیم و آن را با یک مقدار تصادفی دیگر جایگزین می‌کنیم.



شکل ۳: عمل crossover مربوط به الگوریتم ژنتیک موجود در GSOKHM

۴- آزمایش‌ها و نتایج

به منظور سنجش روش پیشنهادی از ۴ مجموعه داده‌ی واقعی که شامل Iris، Wine، Glass و Contraceptive Method Choice (CMC) می‌باشند، استفاده می‌نماییم که در ابعاد کم، متوسط و زیاد هستند. این مجموعه داده‌ها در [۱۵] موجود می باشند. جدول (۱) خلاصه‌ای از ویژگی‌های این مجموعه‌ها را نشان می‌دهد. همچنین در جدول (۲) مقادیر پارامترهایی که در الگوریتم مورد استفاده قرار گرفته، نشان داده شده است.

جدول ۱: ویژگی‌های مجموعه داده‌ها

Name of data set	No. of classes	No. of features	Size of dataset (size of classes in parentheses)
Iris	3	4	150 (50,50,50)
Glass	6	9	214 (70,17,76,13,9,29)
CMC	3	9	1473 (629,334,510)
Wine	3	13	178 (59,71,48)

جدول۲: پارامترهای الگوریتم‌های GSOKHM

Parameter	Value
Psize	18
ω	0.73

C_1	1.5
C_2	1.5
$P_{mutation}$	0.02
$P_{crossover}$	0.5
IterCount	5

۴-۱- نتایج آزمایش‌ها

در این قسمت ما کارایی روش‌های KHM، PSOKHM و GSOKHM را با در نظر گرفتن تابع هدف الگوریتم KHM ارزیابی و مقایسه می‌نماییم. همچنین کیفیت خوشه‌بندی مورد نظر توسط دو معیار زیر بررسی می‌گردد:

مجموعی بر روی تمامی نقاط داده‌ای بر اساس میانگین هارمونیک فاصله از یک نقطه تا تمامی مراکز، همان طور که در معادله (۲-۱۰) آمده است. واضح است که هر چه مقدار این مجموع کوچکتر باشد خوشه‌بندی با کیفیت بهتری خواهیم داشت. معیار F-Measure که از ایده‌های دقت و فراخوانی به منظور بازیابی اطلاعات استفاده می‌کند. هر کلاس i (همان طور که توسط برچسب‌های کلاس در مجموعه داده مورد ارزیابی آمده است) به صورت مجموعی از n_i بخش مطلوب برای یک تحقیق و پرس و جو در نظر گرفته می‌شود. هر خوشه j (که توسط الگوریتم تولید شده) به صورت مجموعی از n_j بخش بازیابی شده توسط یک پرس و جو در نظر گرفته می‌شود. n_{ij} تعداد عناصر کلاس i در داخل خوشه j را می‌دهد. برای هر کلاس i و خوشه j معیارهای دقت و فراخوانی به صورت زیر تعریف می‌شود:

$$r(i,j)=\frac{n_{ij}}{n_i} \quad (۷)$$

$$p(i,j)=\frac{n_{ij}}{n_j} \quad (۸)$$

مقدار F-Measure متناظر نیز به صورت زیر محاسبه می‌گردد:

$$F(i,j)=\frac{(b^2+1).p(i,j).r(i,j)}{b^2.p(i,j)+r(i,j)} \quad (۹)$$

که ما در اینجا مقدار b=1 در نظر می‌گیریم که توازن برابری برای p(i,j) و r(i,j) داشته باشیم. مقدار F-Measure کلی برای مجموعه داده‌های با اندازه‌ی n به صورت زیر می‌باشد:

$$F=\sum_i \frac{n_i}{n}\max_j\{F(i,j)\} \quad (۱۰)$$

واضح است که هر چه مقدار F-Measure بیشتر باشد، کیفیت خوشه‌بندی بهتر خواهد بود.

نتایج آزمایش، میانگین ۱۰ بار اجرای برنامه می‌باشد. الگوریتم های پیشنهادی با استفاده از نرم‌افزار MATLAB 7.6.0 (R2008a) بر روی سیستم عامل Vista Home Primium و بر روی سیستمی با CPU 2.4 GHz و 6 GB RAM انجام گرفته است. آزمایشاتی که تاکنون بر روی الگوریتم KHM انجام شده نشانگر این امر می‌باشد که p یک پارامتر کلیدی برای رسیدن به مقادیر تابع هدف مناسب می‌باشد. بدین منظور ما نیز آزمایشات خود را بر روی مقادیر مختلف p انجام داده و نتایج را در قالب جدول‌هایی با یکدیگر مقایسه کرده‌ایم. جداول موجود حاصل اجرای تابع هدف KHM(X,C) به ازای مقادیر مختلف p به صورت p=2، p=2.5 و p=3 می‌باشد. همچنین علاوه بر استفاده از توابع هدف KHM(X,C) و F-Measure، زمان اجرای الگوریتم‌های پیشنهادی را نیز به عنوان معیار بعدی در ارزیابی الگوریتم‌های گفته شده محاسبه کرده و به نتایج موجود در جدول اضافه کردیم. نهایتا میانگین ۱۰ بار اجرای مستقل الگوریتم‌ها را به عنوان نتایج اصلی ارزیابی در جداول آورده و به مقایسه آنها می‌پردازیم.

جدول3: نتایج حاصل به ازای p=2

	Iris	Glass	Wine	CMC
KHM				
F-Measure	0.8923	0.4831	0.6900	0.4491
KHM (X,C)	74.95	376.33	7,479,216	150,950
Runtime(sec)	0.1811	0.3244	0.2406	0.7720
PSOKHM				
F-Measure	0.8990	0.4245	0.7023	0.4436
KHM (X,C)	58.14	118.80	71,092	49,163
Runtime(sec)	2.19	5.43	2.55	18.68
GSOKHM				
F-Measure	0.9129	0.4354	0.7090	0.4510
KHM (X,C)	11.72	105.25	64,490	9,424
Runtime(sec)	2.73	6.89	3.48	21.45

جدول۴: نتایج حاصل به ازای p=2.5

	Iris	Glass	Wine	CMC
KHM				
F-Measure	0.8853	0.4130	0.6694	0.4496
KHM (X,C)	44.07	633.40	194,607,300	687,737.3
Runtime(sec)	0.1331	0.2898	0.1554	1.5656
PSOKHM				
F-Measure	0.8951	0.4180	0.6835	0.4447
KHM (X,C)	23.159	89.98	8,442,950	82,307.2
Runtime(sec)	2.19	5.88	2.78	19.45
GSOKHM				
F-Measure	0.9017	0.4100	0.6902	0.4446
KHM (X,C)	3.687	70.89	3,572,228	12,921

Runtime(sec)	2.93	7.27	3.71	22.45

جدول۵: نتایج حاصل به ازای p=3

	Iris	Glass	Wine	CMC
KHM				
F-Measure	0.8853	0.4130	0.6694	0.4496
KHM (X,C)	44.07	633.40	194,607,300	687,737.3
Runtime(sec)	0.1331	0.2898	0.1554	1.5656
PSOKHM				
F-Measure	0.8951	0.4180	0.6835	0.4447
KHM (X,C)	23.159	89.98	8,442,950	82,307.2
Runtime(sec)	2.19	5.88	2.78	19.45
GSOKHM				
F-Measure	0.9017	0.4100	0.6902	0.4446
KHM (X,C)	3.687	70.89	3,572,228	12,921
Runtime(sec)	2.93	7.27	3.71	22.45

نتایج حاصل حاکی از این امر می‌باشد که به ازای تمامی مقادیر p مقدار میانگین تابع KHM(X,C) در الگوریتم پیشنهادی GSOKHM از دو الگوریتم KHM و PSOKHM کمتر بوده و در نتیجه به مقدار بهینه‌تری دست یافتیم. از طرفی با بررسی مقدار میانگین F-Measure و مقایسه نتایج حاصل با هم، به این نتیجه دست یافتیم که به جز در مورد داده‌ی CMC، در بقیه موارد این مقدار در GSOKHM بیشتر از دو نمونه‌ی پیشین است و بنابراین از این جهت نیز کارایی بیشتری حاصل شده است. از لحاظ زمان اجرا، این الگوریتم مدت زمان بیشتری را نسبت به KHM نیازمند است ولی با الگوریتم ترکیبی PSOKHM از این نظر قابل مقایسه است.

نهایتا می‌توان نتیجه گرفت که در الگوریتم GSOKHM به دلیل کاهش قابل توجه مقدار تابع KHM(X,C) و افزایش بیشتر F-Measure، کیفیت خوشه‌بندی حاصل از نمونه‌های پیشین به مراتب بهتر خواهد بود.

۵- خلاصه

این مقاله، به بررسی الگوریتم ترکیبی PSOKHM پرداخته است. ایده‌ی اصلی در این الگوریتم بر مبنای بهره‌مندی از مزایای هر دو الگوریتم PSO و KHM می‌باشد و در واقع این ترکیب، هم سرعت همگرایی الگوریتم PSO را بهبود می‌بخشد و هم KHM را از افتادن در دام بهینه محلی حفظ می‌کند. روش پیشنهادی در این مقاله که GSOKHM می‌باشد با همراه کردن الگوریتم تکاملی ژنتیک با PSO، بر روی الگوریتم ترکیبی PSOKHM انجام گرفته است. برای انجام آزمایش از چهار مجموعه داده واقعی استفاده شده است. این الگوریتم‌ها مراکز خوشه‌های داده را با استفاده از مجموعی بر روی تمامی نقاط داده‌ای بر اساس میانگین هارمونیک فاصله از یک نقطه تا تمامی مراکز محاسبه می‌کنند. بر این اساس، این روش نتایج بهتری نسبت به KHM و PSOKHM دارا بود. همچنین از نظر معیار F-Measure نیز شاهد نتایج مساعدتری بودیم.

البته با توجه به این‌که این الگوریتم در امر خوشه‌بندی بسیار کارا می‌باشد ولی از آنجایی که زمان اجرای آن به مراتب بیشتر از KHM می‌باشد، بنابراین در مواقعی که زمان نقش حیاتی و بحرانی در سیستم ایفا می‌کند، نمی‌توان از این روش بهره برد.

مراجع

[1] Yang, F., Sun, T., and Zhang, C., "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization", *Expert Systems with Applications*, 36(9) 847-852, 2009.

[2] Cui, X., and PotokT. E., "Document clustering using Particle Swarm Optimization", *IEEE swarm intelligence symposium*, Pasadena, california, 2005.

[3] Güngör, Z., and Ünler, A., "K-harmonic means data clustering with tabu-search method", *Applied Mathematical Modelling*, 32, 1115–1125, 2008.

[4] Hu, G., Zhou, S., Guan, J., and Hu, X., "Towards effective document clustering: A constrained K-means based approach", *Information Processing and Management*, 44(4), 1397–1409, 2008.

[5] Hammerly, G., and Elkan, C., "Alternatives to the k-means algorithm that find better clusterings", *Proceedings of the 11th international conference on information and knowledge management*, pp. 600–607, 2002.

[6] Liu, B., Wang, L., and Jin, Y. H., "An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers". *Computers and Operations Research*, 35(9), 2791–2806, 2008.

[7] Maitra, M., and Chatterjee, A., "A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding", *Expert Systems with Applications*, 34, 1341–1350, 2008.

[8] Pan, H., Wang, L., and Liu, B., "Particle swarm optimization for function optimization in noisy environment", *Applied Mathematics and Computation*, 181,908–919, 2006.

[9] Tan, P. N., Steinbach, M., and Kumar, V., "Introduction to data mining", pp. 487– 559, Boston: Addison-Wesley, 2005.

[10] Tjhi, W. C., and Chen, L. H., "A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data", *Fuzzy Sets and Systems*, 159(4),371–389, 2008.

[11] Ünler, A., and Güngör, Z., "Applying K-harmonic means clustering to the partmachine classification problem". *Expert Systems with Applications*, pp. 361–406, 2008.

[12] Zhang, B., Hsu, M., and Dayal, U., "K-harmonic means – a data clustering algorithm", *Technical Report HPL-1999-124*. Hewlett-Packard Laboratories, 1999.

[13] Zhang, B., Hsu, M., and Dayal, U., "K-harmonic means". *International workshop on temporal*, spatial and spatio-temporal data mining, TSDM2000. Lyon, France, September 12, 2000.

[14] Zhou, H., and Liu, Y. H., "Accurate integration of multi-view range images using k-means clustering". Pattern Recognition, 41(1), 152–175, 2008.

[15] ftp://ftp.ics.uci.edu./pub/machine-learning-databases.