
# یک روش ترکیبی خوشه‌بندی مبتنی بر الگوریتم‌های رقابت استعماری و C-میانگین فازی

امین گلزاری اسکوئی[۱] ، مهدی هاشم‌زاده[۲]

[۱] دانشجوی کارشناسی ارشد، دانشکده فناوری اطلاعات و مهندسی کامپیوتر دانشگاه شهید مدنی آذربایجان – تبریز – ایران
a.golzari@azaruniv.ac.ir

[۲] استادیار، دانشکده فناوری اطلاعات و مهندسی کامپیوتر دانشگاه شهید مدنی آذربایجان – تبریز – ایران
hashemzadeh@azaruniv.ac.ir

## چکیده

**مساله خوشه‌بندی به منظور کمینه کردن مجموع مجذور انحراف، یک مساله غیر خطی و غیر محدب بوده و دارای تعداد زیادی نقاط بهینه محلی است. هدف از این مقاله، ارائه روشی ترکیبی با استفاده از الگوریتم رقابت استعماری و C-میانگین فازی برای خروج از نقاط بهینه محلی است. استفاده از الگوریتم‌های فراابتکاری برای خروج از نقاط بهینه محلی، توسط محققین بسیاری انجام شده است. تفاوت این روش با سایر روش‌ها، در یافتن بهترین نقاط اولیه است، به طوری که بجای اینکه نقاط اولیه از میان نمونه مجموعه داده انتخاب شود، از بهترین نقاط ممکن که احتمال دارد این نقاط غیر از نقاط مجموعه داده باشد، انتخاب می‌شود. آزمایشات نشان داده است که با انتخاب این نقاط به عنوان نقاط اولیه در الگوریتم C-میانگین فازی ، الگوریتم در کمترین تعداد تکرار همگرا می‌شود و بهترین نتایج را به دنبال دارد. الگوریتم پیشنهادی، بر روی مجموعه داده‌های استاندارد آزمایش شده است. مقایسه نتایج بدست آمده با سایر روش‌ها، به ازای تعداد امپراطوری‌های متفاوت، نشان می‌دهد الگوریتم پیشنهادی کارایی مناسبی را دارد.**

## کلمات کلیدی

**C-میانگین فازی ، خوشه‌بندی رقابت استعماری، بهینه‌سازی، تئوری فازی**

## ۱- مقدمه

خوشه بندی در فضای n بعدی اقلیدسی، فرایند تقسیم بندی یک مجموعه داده یا نمونه به تعداد K گروه یا خوشه براساس شباهت یا عدم شباهت آنها است. در واقع نمونه‌هایی که در یک زیر مجموعه قرار می‌گیرند، بهم شبیه‌اند و با آنهایی که در زیر مجموعه دیگری قرار می‌گیرند متفاوت و غیر شبیه هستند. در بعضی از مسائل خوشه بندی، تعداد K یا خوشه‌ها از پیش تعیین شده است. یکی از رایج‌ترین روش های افرازبندی[۱] برای خوشه بندی، الگوریتم K means است [2] مشکل K means این است که در نقاط بهینه محلی گیر می‌افتد. جواب‌های بدست آمده از این الگوریتم به نقاط یا مراکز خوشه اولیه انتخاب شده، وابسته است. یعنی اگر مراکز مناسبی برای خوشه‌های اولیه انتخاب شده باشد، خوشه بندی خوبی بدست می‌آید. در الگوریتم K-Means تعداد خوشه‌ها یا K از پیش تعریف شده است. در این الگوریتم ابتدا k نمونه[۲] بعنوان مراکز خوشه[۳] بصورت تصادفی انتخاب می‌شود. سپس سایر نمونه‌ها بر اساس کمترین فاصله (معمولا فاصله اقلیدسی) با مرکز خوشه‌ها در یکی از خوشه‌ها قرار می‌گیرند. به این ترتیب K خوشه خواهیم داشت که هر یک حاوی تعدادی نمونه است. سپس میانگین نمونه‌های هر خوشه را محاسبه کرده و بعنوان مراکز جدید آن خوشه در نظر می‌گیریم و براساس آن مرکز خوشه جدید، مجدداً نمونه‌ها خوشه بندی می‌شوند، بدین ترتیب خوشه‌های جدید با نمونه‌های جدید خواهیم داشت. عموما برای خاتمه الگوریتم رابطه (۱) استفاده می‌شود. تلاش الگوریتم بر آنست که میزان این انحراف را کمینه کند. در صورتی که در مقدار SSE[۴] بهبودی حاصل نشود، الگوریتم متوقف می‌شود.

$$SSE = \sum_{i=1}^{K} \sum_{p \epsilon C_i} \|p - m_i\|^2 \qquad (۱)$$

در رابطه (۱) P ، نقطه ای است که نشان دهنده یک نمونه است. $m_i$ ، مرکز خوشه Ci و SSE مجذور انحراف برای همه ی نمونه‌هاست. در این مقاله سعی شده با کمک الگوریتم رقابت استعماری به یک خوشه بندی بهینه دست یابیم. بدین ترتیب که کشورهای اولیه، از نقاط تصادفی از مجموع داده انتخاب شده و برای سیاست جذب به مرحله بعد منتقل می‌شوند. برای ارزیابی عملکرد الگوریتم پیشنهادی از مجموعه داده‌های استاندارد استفاده شده است. مقایسه نتایج بدست آمده با سایر روش‌ها، به ازای امپراطوری‌های متفاوت، نشان می‌دهد الگوریتم پیشنهادی بهترین کارایی را دارد.

در بخشی دوم مروری بر مطالعات انجام شده ی مرتبط با موضوع مقاله ارائه می‌شود. در بخش سوم الگوریتم رقابت استعماری به طور خلاصه معرفی می‌شود. در بخش چهارم الگوریتم پیشنهادی مورد بررسی قرار گرفته و در بخش پنجم نتایج بدست آمده از الگوریتم پیشنهادی بر روی داده‌های استاندارد ارائه می‌گردد و در بخش آخر به جمع بندی و نتیجه گیری پرداخته می‌شود.

## ۲- تحقیقات انجام شده مرتبط

در دو دهه اخیر تحقیقات بسیاری بر مبنای الگوریتم‌های فرا ابتکاری انجام شده است تا بتوان احتمال پیدا کردن نقاط بهینه سراسری[۵] را بالا برد. بیشتر این تحقیقات برپایه الگوریتم‌های با سابقه نظیر الگوریتم ژنتیک، الگوریتم PSO، الگوریتم کولونی مورچگان بود است. از الگوریتم رقابت استعماری به ندرت استفاده شده است و دلیل این امر سابقه کمتر این الگوریتم در مقایسه با سایر الگوریتم‌هاست.

در سال ۱۹۹۳ بابو و مرتی[۶] [3] الگوریتم ژنتیکی برای انتخاب مراکز خوشه اولیه در الگوریتم K means ارائه کردند که در آن از نمایش رشته بیتی[۷] استفاده می‌شد. در این الگوریتم از یک عملگر ساده بازترکیبی برای جابجایی مراکز خوشه والدین استفاده می‌شد، بطوریکه این جابجایی بصورت کاملاً تصادفی در فضای مجموعه داده‌ها انجام می‌شد.

در سال ۲۰۰۳ ، در مقاله [4] دو روش برای خوشه بندی داده‌ها با استفاده از PSO ارائه شد که در آن از PSO برای پیدا کردن مراکز خوشه‌ها استفاده می‌شد که تعداد خوشه‌ها توسط کاربر مشخص می‌شد. در این روش از الگوریتم K means برای پیدا کردن ازدهام ذرات استفاده شد. سپس با استفاده از الگوریتم PSO خوشه‌های بوجود آمده از الگوریتم K means، اصلاح شد.

در سال ۲۰۱۰، طاهر نیکنام و بابک امیری در مقاله [5] روشی جدیدی با عنوان FAPSO-ACO–K را ارائه کردند که در آن سعی شده بود مشکل الگوریتم K means برای مجموعه داده‌های غیر خطی و واگرا حل شود. آنها برای این کار از ترکیب سه الگوریتم الگوریتم PSO فازی و الگوریتم کولونی مورچگان و الگوریتم K means استفاده کردند.

برای اولین بار در سال ۲۰۱۱، روش به نام MICA در مقاله [6] منتشر شد که در آن با استفاده از الگوریتم رقابت استعماری که عملگرهای آن تغییر یافته بود و K means، سعی در یافتن بهترین نقاط بهینه اولیه داشت.

در مقاله [7] روشی ارائه شده است که با استفاده از الگوریتم PSO و C-میانگین فازی، مشکل به دام افتادن در نقاط بهینه محلی را تا حدودی حل می‌کند. در مقاله [8] روشی پیشنهادی با استفاده از کلونی مورچگان و الگوریتم کرنل C-میانگین فازی، علاوه بر مشکل نقاط اولیه، مشکل مربوط به داده‌های جدا پذیر غیر خطی را نیز حل می‌کند. شارما و ورای[۸] در [9] یک روش خوشه‌بندی مبتنی بر الگوریتم ژنتیک ارائه داده‌اند که می‌توان برای حل مسائل خوشه‌بندی با تعداد خوشه‌های مشخص از آن استفاده کرد. در سال ۲۰۱۰ جینگ و همکارانش[۹] [9] برای برطرف کردن مشکلات K means الگوریتم ژنتیک کوانتومی[۱۰] ارائه دادند. همچنین در سال ۲۰۱۰ یو و وانگ[۱۱] [11] الگوریتم تحت عنوان QBCA [۱۲] ارائه نمودند که در آن به کمک توابع ریاضی، داده‌ها به تعدادی هیستوگرام تخصیص داده می‌شوند. راحیلا و همکاران در [12] یک روش خوشه‌بندی مبتنی بر الگوریتم ژنتیک پیشنهاد کردند که در آن از قابلیت جست و جوی الگوریتم ژنتیک به منظور تعیین k مرکز خوشه بهره گرفته شده است. در این روش نیز مقدار k یکی از مقادیر ورودی مسئله می‌باشد. یونگ و همکاران در [13] یک پارتیشن در نظر گرفته‌اند که به عنوان یک رشته به طول n کدگذاری می‌شود، که در آن n تعداد نقاط داده است. عنصر i ام از کروموزوم نشان دهنده تعداد خوشه نقطه متناظر متعلق به آن است. [14] مسئله خوشه‌بندی را به افرازهایی تقسیم کرده است و به حل آن با الگوریتم PSO ترکیبی پرداخته‌است. کرومر و همکاران[۱۳] در [15] روشی را رائه کرده‌اند که در آن با تغییر متغییرهای الگوریتم ژنتیک، نقاط اولیه مناسب را برای خوشه‌بندی پیدا کرده‌اند.

## ۳- الگوریتم رقابت استعماری

روشی در حوزه محاسبات تکاملی است که به یافتن پاسخ بهینه مسائل مختلف بهینه سازی می‌پردازد. این الگوریتم با مدل سازی ریاضی فرایند تکامل اجتماعی - سیاسی، الگوریتمی برای حل مسائل ریاضی بهینه سازی ارائه می‌دهد. از لحاظ کاربرد، این الگوریتم در دسته الگوریتم‌های بهینه سازی تکاملی مانند الگوریتم‌های ژنتیک، روشی بهینه سازی ازدحام ذرات، الگوریتم کلونی مورچگان، الگوریتم تبرید شبیه سازی شده و ... قرار گیرد. همانند همه الگوریتم‌های قرار گرفته در این دسته، الگوریتم رقابت استعماری نیز مجموعه اولیه ای از جواب‌های احتمالی را تشکیل می‌دهد. این جواب‌های اولیه در الگوریتم ژنتیک با عنوان "کروموزوم"، در الگوریتم ازدحام ذرات با عنوان "ذره" و در الگوریتم رقابت استعماری نیز با عنوان "کشور" شناخته می‌شوند. الگوریتم رقابت استعماری با روند خاصی، این جواب‌های اولیه (کشورها) را به تدریج بهبود داده و در نهایت جواب مناسب مسئله بهینه سازی (کشور مطلوب) را در اختیار می‌گذارد[16]. پایه‌های اصلی این الگوریتم را سیاست همسان سازی، رقابت استعماری و انقلاب تشکیل می‌دهند. این الگوریتم با تقلید از روند تکامل اجتماعی، اقتصادی و سیاسی کشورها و با مدل سازی ریاضی بخشهایی از این فرآیند، عملگرهایی را در قالب منظم به صورت الگوریتم ارائه می‌دهد که می‌توانند به حل مسائل پیچیده بهینه سازی کمک کنند [17]. در واقع این الگوریتم جواب‌های مسئله بهینه سازی را در قالب کشورها نگریسته و سعی می‌کند در طی فرآیندی تکرار

شونده این جواب‌ها را رفته رفته بهبود داده و در نهایت به جواب بهینه مسئله برساند[16].

## ۴- الگوریتم پیشنهادی

در روش پیشنهادی، از الگوریتم رقابت استعماری [16]، به عنوان یک الگوریتم فرا ابتکاری برای پیدا کردن مراکز خوشه‌ها استفاده شده است. در این بخش به تشریح اجزای الگوریتم می‌پردازیم.

### ۴-۱- شکل دهی امپراطوری های اولیه

الگوریتم پیشنهادی با مجموعه ای از کشورها ($N_{country}$) که هر کدام نشان دهنده یک جواب برای مسئله مفروض هستند، تحت عنوان جمعیت اولیه شروع به کار می‌کند. نکته مهم در تولید جمعیت اولیه در روش پیشنهادی این است که هر کشور بصورت یک رشته بطول m است که در آن مقدار m برابر با تعداد مراکز خوشه ضرب در ابعاد خواهد بود. شکل زیر یک نمونه ازجمعیت اولیه را نشان می‌دهد که در آن منظور از مرکز خوشه، نقاط تصادفی است که از مجموع داده انتخاب می‌شود.

| مرکز خوشه اول | مرکز خوشه دوم | ... | مرکز خوشه k ام |
|---|---|---|---|

$N_{imp}$تا از بهترین اعضای این جمعیت (کشورهای دارای کمترین مقدار SSE) را به عنوان امپریالیست انتخاب می‌کنیم. باقیمانده $N_{col}$ تا از کشورها، مستعمراتی را تشکیل می‌دهند که هرکدام به یک امپراطوری تعلق دارند. برای تقسیم مستعمرات اولیه بین امپریالست‌ها، به هر امپریالیست، تعدادی از مستعمرات را که این تعداد، متناسب با قدرت آن است، می‌دهیم. قدرت نسبی نرمالیزه‌ی هر امپریالیست، به صورت (۲) محاسبه شده و بر مبنای آن، کشورهای مستعمره، بین امپریالسیت‌ها تقسیم می‌شوند. که در آن $c_n$ ، هزینه امپریالست nام، $\max\{C_i\}$ بیشترین هزینه میان امپریالیست‌ها و $C_n$ ، هزینه نرمالیزه شده این امپریالیست، می‌باشد. هر امپریالیستی که دارای هزینه بیشتری باشد (امپریالیست ضعیفتری باشد)، دارای هزینه نرمالیزه کمتری خواهد بود(۳).

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (۲)$$

$$C_n = \max\{C_i\} - C_n \quad (۳)$$

از یک دید دیگر، قدرت نرمالیزه شده یک امپریالیست، نسبت مستعمراتی است که توسط آن امپریالیست اداره می‌شود. بنابراین تعداد اولیه‌ی مستعمرات یک امپریالیست برابر خواهد بود با:

$$N.C_n = \text{round}\{p_n . (N_{col})\} \quad (۴)$$

که در آن $N.C_n$، تعداد اولیه مستعمرات یک امپراطوری و $N_{col}$ نیز تعداد کل کشورهای مستعمره موجود در جمعیت کشورهای اولیه است.

round نیز تابعی است که نزدیک‌ترین عدد صحیح به یک عدد اعشاری را می‌دهد. با در نظر گرفتن $N.C$ برای هر امپراطوری، به این تعداد از کشورهای مستعمره اولیه را به صورت تصادفی انتخاب کرده و به امپریالیست nام می‌دهیم.

### ۴-۲- مدل سازی سیاست جذب[۴]: حرکت مستمره‌ها به سمت امپریالیست‌ها

براساس شکل (۱)، کشور مستعمره (Colony)، به اندازه x واحد درجهت خط واصل مستعمره به استعمارگر (Imperialist)، حرکت کرده و به موقعیت جدید (New Position of Colony)، کشانده می‌شود. در این شکل، فاصله میان استعمارگر و مستعمره با d نشان داده شده است. x نیز عددی تصادفی می‌باشد. یعنی برای x داریم (۵):

$$x \in U(0 . \beta \times d) \quad (۵)$$

که در آن β عددی بزرگتر از یک و نزدیک به ۲ می‌باشد. یک انتخاب مناسب می‌تواند β=2 باشد. وجود ضریب β>1 باعث می‌شود تا کشور مستعمره در حین حرکت به سمت کشور استعمارگر، از جهت‌های مختلف به آن نزدیک شود. در واقع β یک پارمتر کنترلی است.

کشور مستمره دقیقا روی خط واصل حرکت نمی‌کند بلکه با یک زاویه از موقعیت اولیه منحرف می‌شود. این زاویه بصورت تصادفی و با یک توزیع یکنواخت در نظر می‌گیریم (۶).

$$\theta \in U(-\gamma . \gamma) \quad (۶)$$

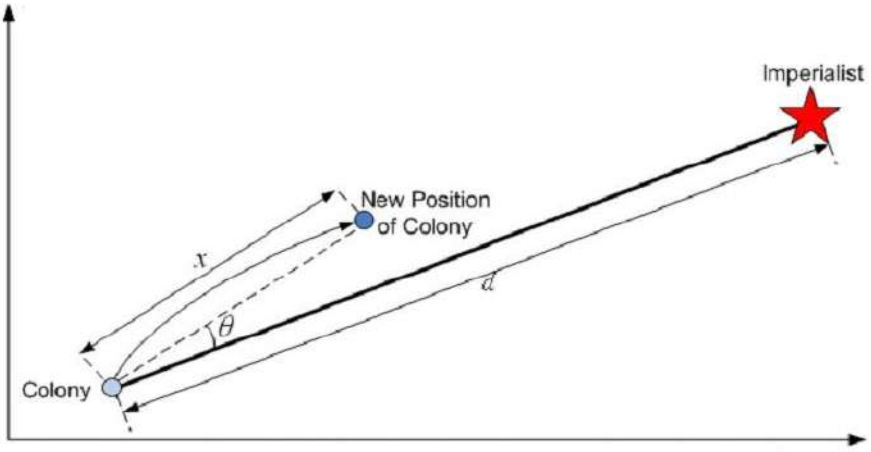
در این رابطه، $\gamma$ پارامتری دلخواه می‌باشد که افزایش آن باعث افزایش جستجوی اطراف امپریالیست شده و کاهش آن نیز باعث می‌شود تا مستعمرات تا حد ممکن، به بردار واصل مستعمره به استعمارگر، نزدیک حرکت کنند. با در نظر گرفتن واحد رادیان برای θ، عددی نزدیک به π/4، در اکثر پیاده‌سازی‌ها، انتخاب مناسبی بوده است. در نهایت واحد کاندید جدید بصورت زیر تولید می‌شود:

$$\{x\}_{new} = \{x\}_{old} + U(0 \ \beta \times d) \times \{V_1\} \quad (۷)$$

که در فرمول فوق $V_1$ برداری است که نقطه شروع آن مکان قبلی مستعمره و جهت آن به سمت امپراطوری است.

### ۴-۳- جابجایی موقعیت مستعمره و امپریالیست

در حین حرکت مستعمرات به سمت کشور استعمارگر، ممکن است بعضی از این مستعمرات به موقعیتی بهتر از امپریالیست برسند (به نقاطی در تابع هزینه برسند که هزینه کمتری را نسبت به مقدار تابع




**شکل (۱) حرکت کشور مستعمره به سمت امپریالیست [16]**

هزینه در موقعیت امپریالیست، تولید می‌کنند). در این حالت، کشور استعمارگر و کشور مستعمره، جای خود را با همدیگر عوض کرده و الگوریتم با کشور استعمارگر در موقعیت جدید ادامه یافته و این بار این کشور امپریالیست جدید است که شروع به اعمال سیاست همگون‌سازی بر مستعمرات خود می‌کند.

## ۴-۴- قدرت کل یک امپراطوری

قدرت یک امپراطوری برابر است با قدرت کشور استعمارگر، به اضافه درصدی از قدرت کل مستعمرات آن. بدین ترتیب برای هزینه کل یک امپراطوری داریم (۸):

$$T.C_n = Cost(imperialist) + \xi\, mean\{Cost(colones\ of\ empire_n)\} \quad (۸)$$

که در آن $T.C_n$ هزینه کل امپراطوری nام و $\xi$ عددی مثبت است که معمولاً بین صفر و یک و نزدیک به صفر در نظر گرفته می‌شود. کوچک در نظر گرفتن $\xi$ ، باعث می‌شود که هزینه کل یک امپراطوری، تقریباً برابر با هزینه حکومت مرکزی آن (کشور امپریالیست)، شود و افزایش $\xi$ نیز باعث افزایش تاثیر میزان هزینه مستعمرات یک امپراطوری در تعیین هزینه کل آن می‌شود. در حالت نوعی $\xi = 0.05$ در اکثر پیاده‌سازی به جوابهای مطلوبی منجر شده است.

## ۴-۵- رقابت استعماری

به مرور زمان، امپراطوری‌های ضعیف، مستعمرات خود را از دست داده و امپراطوری‌های قویتر، این مستعمرات را تصاحب کرده و بر قدرت خویش می‌افزایند. برای مدل کردن این واقعیت، فرض می‌کنیم که امپراطوری در حال حذف، ضعیف‌ترین امپراطوری موجود است. بدین ترتیب، در تکرار الگوریتم، یکی یا چند تا از ضعیف‌ترین مستعمرات ضعیف‌ترین امپراطوری را برداشته و برای تصاحب این مستعمرات، رقابتی را میان کلیه امپراطوری‌ها ایجاد می‌کنیم. مستعمرات مذکور، لزوماً توسط قویترین امپراطوری، تصاحب نخواهند شد، بلکه امپراطوری‌های قویتر، احتمال تصاحب بیشتری دارند. برای مدل‌سازی رقابت میان امپراطوری‌ها برای تصاحب این مستعمرات، ابتدا احتمال تصاحب هر امپراطوری (که متناسب با قدرت آن امپراطوری می‌باشد)، را با در نظر گرفتن هزینه کل امپراطوری، به ترتیب زیر محاسبه می‌کنیم. ابتدا از روی هزینه کل امپراطوری، هزینه کل نرمالیزه شده آن را تعیین می‌کنیم (۹):

$$N.T.C_n = \max\{T.C_i\} - T.C_n \quad (۹)$$

در این رابطه $T.C_n$، هزینه کل امپراطوری nام و $N.T.C_n$ نیز، هزینه کل نرمالیزه شده آن امپراطوری می‌باشد. هر امپراطوری که $T.C_n$ کمتری داشته باشد $N.T.C_n$ بیشتری خواهد داشت. در حقیقت $T.C_n$ معادل هزینه کل یک امپراطوری و $N.T.C_n$ معادل قدرت کل آن می باشد. امپراطوری با کمترین هزینه، دارای بیشترین قدرت است. با داشتن هزینه کل نرمالیزه شده، احتمال (قدرت) تصاحب مستعمره رقابت، توسط هر امپراطوری، به صورت زیر محاسبه می‌شود (۱۰).

$$P_{P_n} = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i} \right| \quad (۱۰)$$

با داشتن احتمال تصاحب هر امپراطوری، برای اینکه مستعمرات مذکور را به صورت تصادفی، ولی با احتمال وابسته به احتمال تصاحب هر امپراطوری، بین امپراطوری‌ها تقسیم کنیم؛ بردار $P$را از روی مقادیر احتمال فوق، به صورت زیر تشکیل میدهیم (۱۱).

$$P = [P_{P_1}, P_{P_2}, \dots, P_{P_{N_{imp}}}] \quad (۱۱)$$

بردار $P$ دارای سایز $1*N_{imp}$ می‌باشد و از مقادیر احتمال تصاحب امپراطوری‌ها تشکیل شده است. سپس بردار تصادفی $R$ ، همسایز با

بردار $P$ را تشکیل می‌دهیم. آرایه‌های این بردار، اعدادی تصادفی با توزیع یکنواخت در بازه $[0,1]$ می‌باشند (۱۲).

$$R = [r_1. r_2. \dots . r_{N_{im}}] \quad (۱۲)$$

سپس بردار $D$را به صورت زیر تشکیل می‌دهیم (۱۳).

$$D = P - R = [D_1. D_2. D_3. \dots . D_{N_{imp}}] \quad (۱۳)$$

با داشتن بردار $D$ ، مستعمرات مذکور را به امپراطوری‌ای می‌دهیم که اندیس مربوط به آن در بردار $D$ بزرگتر از بقیه می‌باشد. امپراطوری‌ای که بیشترین احتمال تصاحب را داشته باشد، با احتمال بیشتری اندیس مربوط به آن در بردار $D$ ، بیشترین مقدار را خواهد داشت. با تصاحب مستعمره توسط یکی از امپراطوری ها، عملیات این مرحله از الگوریتم نیز به پایان می‌رسد.

## ۴-۶- سقوط امپراطوری‌های ضعیف

در جریان رقابت‌های امپریالیستی، خواه ناخواه، امپراطوریهای ضعیف به تدریج سقوط کرده و مستعمراتشان به دست امپراطوری‌های قوی‌تر می‌افتد. شروط متفاوتی را می‌توان برای سقوط یک امپراطوری در نظر گرفت. در الگوریتم پیشنهاد شده، یک امپراطوری زمانی حذف شده تلقی می‌شود که مستعمرات خود را از دست داده باشد.

## ۴-۷- همگرایی

در الگوریتم پیشنهادی سه شرط برای پایان الگوریتم مد نظر قرار گرفته است. اگر یکی از این سه شرط برقرار باشد الگوریتم پایان می‌یابد:

۱) تعداد تکرارها به بیشینه مقدار خود برسد این بیشینه مقدار توسط کاربر تعیین می‌شود.

۲) اصلی‌ترین شرط برای متوقف شدن الگوریتم زمانی است که معیار انحراف درون خوشه‌ها در چندین تکرار متوالی با هم برابر باشد.

۳) در نهایت یک امپراطوری باقی بماند.

## ۵- نتایج آزمایشات

در این بخش ارزیابی از الگوریتم پیشنهادی در مقایسه با سایر الگوریتم-ها بر پنج مجموعه داده واقعی شامل Ecoli، Coil2، Iris، Glass، Bupa از انبار داده UCI انجام شد [18]. اطلاعات مربوط به این دیتاست‌ها در جدول (۱) خلاصه شده است.

**جدول (۱) دیتاست انتخاب شده از UCI**

| دیتاست | نمونه | صفت | خوشه |
|---|---|---|---|
| Ecoli | ۳۳۶ | ۸ | ۸ |
| Coil2 | ۲۱۶ | ۱۰۰۰ | ۳ |
| Iris | ۱۵۰ | ۴ | ۳ |
| Bupa | ۳۴۵ | ۷ | ۲ |
| Glass | ۲۱۴ | ۱۰ | ۶ |

پارامترهای مورد آزمایش در نتایج بدست آمده شامل: پارامتر کنترلی، $\beta = 2$، پارامتر تاثیر گزاری مستعمره‌ها، $\xi = 0.05$ ، اندازه جمعیت ۱۰۰ کشور و تعداد کشورهای امپراطوری را برای سه مقدار ۲، ۵، ۱۰ مورد آزمایش قرار دادیم. تعداد مستعمره انتخاب شده برای حذف شدن در هر تکرار را، یک بار برابر ۱ ویک بار برابر ۲ قرار دادیم. در واقع الگوریتم ارائه شده را برای شش حالت مختلفی از انتخاب امپراطور و تعداد مستعمره انتخاب شده برای حذف شدن در هر تکرار، مورد ارزیابی قرار دادیم. مقدار ماکزیمم تکرار را برابر ۱۰۰ قرار دادیم. الگوریتم ارائه شده را با الگوریتم K means و دو نسخه بهبود یافته آن یعنی C-میانگین فازی و Kernel K means و همچنین با روش ارائه شده در مقاله [۱] مقایسه کردیم. هر یک از الگوریتم‌ها ۱۰ بار تکرار شده و مقدار میانگین آن ثبت شده است. درجه فازی سازی در C-میانگین فازی و الگوریتم ارائه شده را برابر ۲ قراردادیم. در الگوریتم Kernel K means از تابع کرنل گاووسی استفاده کردیم که در آن مقادیر سیگما را ۱، ۲ و ۳ قرار دادیم و بهترین حالت ممکن را برای مقایسه با الگوریتم ارائه شده انتخاب کردیم. در آزمایشات از دو معیار SSE و Silhouette برای نمایش کارایی این الگوریتم استفاده کردیم. معیار Silhouette در واقع نشان دهنده کیفیت خوشه بندی است که بستگی به بیشترین شباهت اعضا درون یک خوشه و کمترین شباهت اعضای یک خوشه از اعضای سایر خوشه‌ها است.

با توجه به جدول نتایج چهار دیتاست iris، Ecoli، Coil2 و Bupa مشاهده می‌شود که در هر ۶ حالت معیار Silhouette نسبت به سایر الگوریتم‌ها بیشترین مقدار را داشته است و بهترین نتایج زمانی حاصل شده است که تعداد مستعمره‌های حذف شده برابر یک باشد. SSE برای الگوریتم C-میانگین فازی کمترین مقدار بود است ولی معیار Silhouette مقدار پایینی بوده است که نشان دهنده خوشه بندی بی کیفیت است.

**جدول (۲) مجموع داده Iris**

| | Silhouette coefficient | SSE |
|---|---|---|
| K means | ۰,۵۵۱۰ | ۳۱۰,۶۱۷۱ |
| C-میانگین فازی | ۰,۵۴۹۳ | ۹۶,۹۲۷۶ |
| Kernel K means (σ=2) | ۰,۵۴۰۷ | ۲۹۵,۲۳۲۶ |
| الگوریتم ارائه شده در [۱] | ۰,۷۰۱۱ | ۱۰۵,۰۳۶۵ |
| ۲= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۷۱۵۴ | ۹۸,۰۴۵۴ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۷۱۳۷ | ۹۸,۷۴۴۵ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۷۱۲۳ | ۹۸,۷۸۲۵ |
| ۱= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۷۲۹۷ | ۹۷,۵۵۳۷ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۷۱۹۸ | ۹۹,۱۳۲۹ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۷۱۷۸ | ۹۹,۶۸۲۹ |

**جدول (۳) مجموع داده Ecoli**

| | Silhouette coefficient | SSE |
|---|---|---|
| K means | ۰,۲۳۰۸ | ۱۶۶,۵۲۸۹ |
| C-میانگین فازی | ۰,۲۰۰۳ | ۶۵,۰۱۹۱ |
| Kernel K means (σ=1) | ۰,۲۲۹۷ | ۱۰۸,۲۱۴۷ |
| الگوریتم ارائه شده در [۱] | ۰,۲۶۲۳ | ۷۲,۰۲۱۷ |
| ۲= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۳۲۶۹ | ۶۹,۹۷۷ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۳۱۰۳ | ۷۳,۷۶۶۹ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۳۱۳۸ | ۷۲,۱۱۴۳ |
| ۱= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۳۳۹۵ | ۷۱,۷۳۳۹ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۲۹۱۱ | ۷۱,۹۱۶۳ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۲۴۰۵ | ۷۲,۶۵۰۰ |

**جدول (۴) مجموع داده Glass**

| | Silhouette coefficient | SSE |
|---|---|---|
| K means | ۰,۳۷۸۹ | ۴۸۱,۲۹۷۸ |
| C-میانگین فازی | ۰,۲۴۱۱ | ۲۱۳,۴۷۸۹ |
| Kernel K means (σ=2) | ۰,۲۱۰۸ | ۱۸۲,۵۶۷۰ |
| الگوریتم ارائه شده در [۱] | ۰,۲۴۵۴ | ۲۴۱,۶۶۸۳ |
| ۲= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۲۸۶۷ | ۲۳۸,۲۴۱۱ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۲۶۶۷ | ۲۴۱,۰۸۵۹ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۲۸۹۸ | ۲۳۹,۶۶۶۱ |
| ۱= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۳۴۵۰ | ۲۴۲,۴۹۳۵ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۲۳۲۸ | ۲۴۴,۴۳۸۶ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۲۹۸۴ | ۲۴۰,۷۸۱۳ |

**جدول (۵) مجموع داده Bupa**

| | Silhouette coefficient | SSE |
|---|---|---|
| K means | ۰,۶۳۲۸ | ۱۴۸۲۲ |
| C-میانگین فازی | ۰,۵۸۷۳ | ۹۸۶۵,۴ |

| | | |
|---|---|---|
| Kernel K means (σ=1) | ۰,۲۹۳۰ | ۱۲۱۰۰ |
| الگوریتم ارائه شده در [۱] | ۰,۶۵۳۰ | ۱۰۵۳۱ |
| ۲= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۷۲۶۰ | ۹۸۴۸,۷ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۷۳۸۲ | ۹۸۱۵,۴ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۷۳۴۰ | ۹۷۹۴,۸ |
| ۱= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۷۳۳۹ | ۹۷۹۴,۵ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۷۲۷۸ | ۹۸۲۹,۲ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۷۴۳۶ | ۹۸۵۵,۱ |

**جدول (۶) مجموع داده Coil2**

| | Silhouette coefficient | SSE |
|---|---|---|
| K means | ۰,۱۰۴۷ | ۲۶۷,۵۴۷۶ |
| C-میانگین فازی | ۰,۱۰۴۶ | ۱۹۹,۶۸۵۷ |
| Kernel K means (σ=2) | ۰,۱۱۶۲ | ۲۶۲,۹۴۷۰ |
| الگوریتم ارائه شده در [۱] | ۰,۱۷۱۱ | ۲۴۷,۱۹۰۲ |
| ۲= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۱۷۹۱ | ۲۴۱,۱۱۱۳ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۱۷۶۳ | ۲۴۰,۹۶۱۰ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۱۸۰۰ | ۲۴۰,۳۱۳۱ |
| ۱= مستعمره حذف شده | | |
| الگوریتم ارائه شده ( ۲ = امپراطوری) | ۰,۱۸۵۲ | ۲۳۹,۹۹۳۵ |
| الگوریتم ارائه شده ( ۵ = امپراطوری) | ۰,۱۷۸۷ | ۲۴۱,۲۰۳۲ |
| الگوریتم ارائه شده ( ۱۰ = امپراطوری) | ۰,۱۸۶۹ | ۲۳۹,۶۶۶۸ |

نتایج نشان می‌دهد که الگوریتم پیشنهادی بهترین کارایی را زمانی بدست آورده است که مستعمره‌های حذف شده در هر تکرار، کمترین مقدار باشد. علاوه بر این نکته، با تخصیص دادن بهترین کشور به عنوان مرکز خوشه به الگوریتم C-میانگین فازی ، الگوریتم در کمترین تعداد تکرار همگرا شده و بهترین نتیجه بدست می‌آید.

## ۶- نتیجه گیری

با توجه به اینکه در الگوریتم C-میانگین فازی نمونه‌های اولیه به صورت تصادفی انتخاب می‌شوند، ممکن است الگوریتم در دام بهینه محلی قرار گرفته و جواب بهینه را تولید ننماید. لذا جهت خروج از وضعیت بهینه محلی، با ترکیب الگوریتم فوق با الگوریتم رقابت استعماری مدل خوشه بندی جدیدی ارائه شده است که سبب خروج از دام بهینه محلی و تولید جواب بهتر می‌گردد. تفاوت این روش با سایر الگوریتم‌های فرا ابتکاری خوشه بندی، در یافتن بهترین نقاط اولیه است، به طوری که بجای اینکه نقاط اولیه از میان نمونه مجموعه داده انتخاب شود از بهترین نقاط ممکن، که احتمال دارد این نقاط غیر از نقاط مجموعه داده باشد، انتخاب می‌شود.

الگوریتم پیشنهادی با تعداد مختلفی از امپراطوری‌ها و تعداد مستعمراتی که در هر تکرار حذف می‌شوند، که مجموعا ۶ حالت مختلف را تشکیل می‌دهند، آزمایش شد. نتایج بدست آمده نشان می‌دهد با تعداد امپراطوری کمتر و تعداد کمتر مستعمراتی که در هر تکرار حذف می-شوند، نتایج بهتری نسبت به سایر روش‌ها بدست می‌آید.

## مراجع


۱. یقینی, م., ر. سلطانیان ,ج. نوری, یک روش ترکیبی خوشه بندی مبتنی بر الگوریتم ژنتیک با استفاده از عملگرهای جدید تغییر. فصلنامه بین المللی مهندسی صنایع و مدیریت تولید ۱۳۹۱.

2. Han, J., J. Pei, and M. Kamber, Data mining: *concepts and techniques*. 2011: Elsevier.
3. Babu, G.P. and M.N. Murty, *A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm*. Pattern Recognition Letters, 1993. 14(10): p. 763-769.
4. Van der Merwe, D. and A.P. Engelbrecht. *Data clustering using particle swarm optimization*. in Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. 2003. IEEE.
5. Niknam, T. and B. Amiri, *An efficient hybrid approach based on PSO, ACO and k-means for*

*cluster analysis*. Applied Soft Computing, 2010. 10(1): p. 183-197.

6. Niknam, T., et al., *An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering*. Engineering Applications of Artificial Intelligence, 2011. 24(2): p. 306-317.
7. Kuo, R., et al., *Integration of particle swarm optimization and genetic algorithm for dynamic clustering*. Information Sciences, 2012. 195: p. 124-140.
8. DoğAn, B. and M. Korürek, *A new ECG beat clustering method based on kernelized fuzzy c-means and hybrid ant colony optimization for continuous domains.* Applied Soft Computing, 2012. 12(11): p. 3442-3451.
9. Sharma, S., Rai, Sh., *"Genetic K-Means Algorithm – Implementation and Analysis"*, International Journal of Recent Technology and Engineering (IJRTE), Vol.1, Issue.2, June 2012, pp. 117-120.
10. Xiao, J., et al., *A quantum-inspired genetic algorithm for k-means clustering*. Expert Systems with Applications, 2010. 37(7): p. 4966-4973. of Recent Technology and Engineering.
11. Yu, Z. and H.-S. Wong, *Quantization-based clustering algorithm*. Pattern Recognition, 2010. 43(8): p. 2698-2711
12. Sheikh, R.H., M. Raghuwanshi, and A.N. Jaiswal. *Genetic algorithm based clustering: a survey*. in *Emerging Trends in Engineering and Technology*, 2008 *ICETET'08. First International Conference on*. 2008. IEEE.
13. Liu, Y., et al. *Finding the optimal number of clusters using genetic algorithms*. in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*. 2008. IEEE.
14. Nanda, S. and G. Panda. *Accurate partitional clustering algorithm based on immunized PSO*. in *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*. 2012. IEEE.
15. Krömer, P., J. Platoš, and V. Snášel. *Genetic algorithm for clustering accelerated by the CUDA platform*. in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*. 2012. IEEE
16. Atashpaz-Gargari, E. and C. Lucas. Imperialist competitive algorithm: *an algorithm for optimization inspired by imperialistic competition*. in Evolutionary computation, 2007. CEC 2007. IEEE Congress on. 2007. IEEE.
17. Nazari-Shirkouhi, S., et al., *Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm*. Expert Systems with Applications, 2010. 37(12): p. 7615-7626.
18. www.UCI.com

**زیرنویس‌ها**

۱ Partitioning
۲ Object
۳ Mean value
٤ Sum of squared Euclidean distance
٥ Global optimal
٦ Babo and Murty
۷ Bit string representation
۸ Sharma, S., Rai, Sh
۹ Jing Xiao, YuPing Yan, Jun Zhang and Yong Tang
۱۰ The quantum-inspired genetic algorithm
۱۱ Zhiwen Yu and Hau-San Wong
۱۲ Quantization -Based Clustering Algorithm (QBCA)
۱۳ Krömer, P., J. Platoš, and V. Snášel
۱٤ Assimilation