IDMC'07
20-21 Nov.2007
Amir Kabir University

# تطبیق پارامترهای خوشه‌بندی مورچه‌ای به کمک CLA-PSO

برنا جعفرپور[۱]، محمد رضا میبدی[۲]

## چکیده

**خوشه‌بندی مورچه‌ای روشی الهام گرفته از طبیعت می‌باشد دارای ویژگی های جالبی می‌باشد که باعث توجه محققان با این روش در سالهای اخیر شده است. این روش دارای پارامترهای متعددی از جمله پارامترهای مربوط به برداشتن، گذاشتن داده ها و شعاع دید می‌باشد که تاثیر زیادی در عملکرد و همگرایی الگوریتم دارند و معمولا به صورت آزمایش و خطا تعیین می‌گردند. در این مقاله روشی مبتنی برCLA-PSO که یک مدل گسسته ی PSO می‌باشد برای تطبیق اتوماتیک پارامترهای خوشه‌بندی مورچه‌ای پیشنهاد می‌گردد. به منظور بررسی کارایی روش پیشنهادی، این روش با تنها روش موجود تطبیق پارامترها که بر اساس الگوریتمهای ژنتیکی می‌باشد مقایسه گردیده است. نتایج آزمایشهای انجام گرفته حاکی از کارایی بالای روش پیشنهادی در مقایسه با روش مبتنی بر الگوریتم ژنتیکی و روش خوشه‌بندی k-means می‌باشد.**

**کلمات کلیدی: خوشه‌بندی مورچه‌ای، اتوماتاهای یادگیر، CLA-PSO، تطبیق پارامتر**

# Adaptation of Ant Clustering Parameters using CLA-PSO

Borna Jafarpour, Mohammad Reza Meybodi
Amirkabir University of Technology, Computer Engineering and IT Department,
Soft Computing Laboratory
jafarpour@cic.aut.ac.ir, mmeybodi@aut.ac.ir

## Abstract

**Ant based clustering is a bio-inspired method for clustering. This method has some interesting characteristic that has prompted special attention form the researchers in the past few years. This technique has many parameters to tune, such as drop, pick, and radius of perception parameters that have crucial role in performance and convergence of the algorithm. These parameters are usually tuned with trial and error. In this paper we use CLA-PSO, which is a discrete version of PSO based on cellular learning automata, to tune parameters of ant clustering technique. To asses the performance of the proposed method, we compared the proposed method with the only reported method for automatic adaptation of ant clustering parameters which is based on genetic algorithm. Experimental result shows the superiority of the proposed method over the method based on genetic algorithm and k-means method.**

**Key words: Ant based Clustering, Learning Automata, CLA-PSO, Parameter Adaptation**

---

[1] دانشجوی هوش مصنوعی و رباتیک، دانشگاه صنعتی امیر کبیر، دانشکده مهندسی کامپیوتر و فن آوری اطلاعات، آزمایشگاه محاسبات نرم
jafarpour@cic.aut.ac.ir
[2] استاد تمام، دانشگاه صنعتی امیر کبیر، دانشکده مهندسی کامپیوتر و فن آوری اطلاعات، آزمایشگاه محاسبات نرم mmeybodi@aut.ac.ir

**۱. مقدمه**

روش خوشه‌بندی مورچه‌ای که الهام گرفته از رفتار مورچه ها در طبیعت می‌باشد اولین بار در [3] برای مجموعه ای از ربات ها که اشیاء را دسته بندی می‌کردند ارائه شد. این الگوریتم در [14] برای خوشه‌بندی داده ها تصحیح و ارائه شد. در روش خوشه‌بندی مورچه‌ای از تعدادی مورچه با عملکرد ساده و بدون کنترل مرکزی استفاده می‌شود. در این روش یک شبکه ۲ بعدی از سلولها وجود دارد که داده‌ها و مورچه‌ها برروی سلولهای آن قرار دارند و مورچه‌ها می‌توانند بر روی آنها حرکت کنند. در این روش بر خلاف بسیاری از روش‌های خوشه بندی احتیاج به از پیش تعیین کردن تعداد خوشه‌ها نمی‌باشد و تعداد آنها بطور خودکار در حین فرایند خوشه بندی تعیین میگردد. یکی دیگر از ویژگی های مهم این روش اینست که خوشه‌بندی در یک فضای ۲ بعدی به جای فضای n بعدی انجام می‌شود که امکان ارزیابی خوشه‌بندی توسط انسان را فراهم می‌کند. این روش دارای کاربردهای زیادی از جمله خوشه‌بندی مستندات [6][7] ، قسمت بندی[۱] گراف [9][11] و قسمت بندی در VLSI [10] و غیره می‌باشد.

الگوریتم ژنتیک[۲][4] روشی الهام گرفته از طبیعت برای جستجو در مسائل با فضای حالت بزرگ می‌باشد و پر پایه ی قانون بقای شایسته ترین[۳] داروین عمل می‌کند. در این روش مجموعه ای از افراد وجود دارند که هر کدام نشان دهنده ی یک جواب از مسئله می‌باشند و طی نسلهای الگوریتم بهبود داده می‌شوند. این روش تعدادی عملگر دارد که عملگرهای ژنتیکی واقعی را شبیه سازی می‌کند : انتخاب[۴]، بازترکیبی[۵] و جهش[۶]. عمگر انتخاب باعث انتخاب بهترین افراد و انتقال آنها به نسل بعد می‌شود. عملگر بازترکیبی برای تولید فرزندان از دو فرد (والدین) به کار می‌رود. این عملگر با این امید به کار می‌رود که فرزندان خصوصیات خوب والدین را به ارث ببرند و بهتر از والدین بشوند. جهش در الگوریتم ژنتیک، همانند جهش در طبیعت باعث تنوع افرادجمعیت می‌شود. جهش در الگوریتم ژنتیک مانع از همگرایی زودرس الگوریتم شده و باعث تولید جوابهای بهتر می‌شود.

PSO [۷] روشی است که اولین بار برای بهینه سازی غیر خطی توابع پیوسته ارائه شده است[12]. این روش با کمک شبیه سازی مدلهای اجتماعی بدست آمده است. این الگوریتم ریشه در دو مدل کلی دارد: ۱.حیات مصنوعی [۸] مانند مدل های گروه پرندگان و دسته ماهی ها و تئوری آشوب [۹] ۲.مدل های تکاملی به ویژه الگوریتم ژنتیک و برنامه ریزی تکاملی [۱۰]. PSO شباهت زیادی به دیگر مدلهای تکاملی مانند الگوریتم ژنتیک دارد به عنوان مثال این روش با مجموعه ای از افراد اولیه شروع به کار می‌کند و سعی می‌کند که افراد اولیه را در طول زمان بهبود ببخشد. تفاوت در اینجاست که PSO همانند الگوریتم ژنتیک عملگرهایی مانند بازترکیبی و جهش ندارد. در این روش جواب ها ، که ذره[۱۱] نامیده می‌شوند در فضای حالت به دنبال افراد بهتر جمعیت حرکت می‌کنند. در این روش افراد جمعیت دارای حافظه می‌باشند و بهترین جوابی را که تا کنون یافته اند به خاطر می‌سپارند. افراد در فضای حالت مسئله ، به سمت بهترین جواب یافته شده توسط خودشان و کل جمعیت حرکت می‌کنند. دو نوع PSO از نظر فضای حالت معرفی شده است پیوسته [12] و گسسته[8][13][16] .در مدل پیوسته PSO در یک فضای حالت پیوسته جستجو می‌کند ولی در حالت گسسته فضای حالت گسسته می‌باشد.در [8] یک مدل گسسته ی PSO ارائه شده است که برخلاف سایر مدلهای گسسته ی PSO از همگرایی زودرس رنج نمی‌برد و کارایی بالایی در بهینه سازی دارد.

خوشه بندی مورچه ای دارای پارامترهای متعددی از جمله پارامترهای مربوط به برداشتن، گذاشتن داده ها و شعاع دید می‌باشد که تاثیر زیادی بر عملکرد و همگرایی الگوریتم دارند و معمولا به صورت آزمایش و خطا تعیین می‌گردند. در این مقاله روشی مبتنی بر CLA-PSO[۱۲] [8] ،که یک مدل گسسته ی دودویی PSO می‌باشد، برای تطبیق اتوماتیک پارامترهای خوشه‌بندی مورچه‌ای پیشنهاد می‌گردد. در این مقاله از دو نوع مورچه برای خوشه بندی استفاده کرده ایم که هرکدام نیمی از زمان اجرای خوشه‌بندی را در اختیار دارند و از CLA-PSO برای تطبیق پارامترهای این مورچه‌ها استفاده کرده‌ایم. تنها روش گزارش شده برای تنظیم اتوماتیک پارامترهای خوشه‌بندی مورچه‌ای مبتنی بر الگوریتمهای ژنتیکی می‌باشد که از یک نوع مورچه برای خوشه‌بندی استفاده کرده و پارامترهای آن را تنظیم کرده است و توسط آرانها[۱۳] و همکاران ارائه گردیده است[1]. مقایسه ها، کارایی بالای روش پیشنهادی را در مقایسه با روش موجود و K-means نشان می‌دهد. ادامه ی مقاله به این شکل است : بخش ۲ خوشه‌بندی مورچه‌ای را معرفی می‌کند. در بخش ۳ و ۴ به ترتیب اتوماتای یادگیر سلولی و CLA-PSO توضیح داده می‌شوند. بخش ۵ چگونگی استفاده از CLA-PSO را برای تطبیق پارامترها در خوشه‌بندی مورچه‌ای نشان می‌دهد. مقایسه ها در بخش ۶ و نتیجه گیری در بخش ۷ آورده شده اند.

## ۲. خوشه‌بندی مورچه‌ای

روش خوشه‌بندی مورچه‌ای الهام گرفته از رفتار مورچه ها در طبیعت می‌باشد. مورچه ها در طبیعت می‌توانند اشیاء مختلف را بدون وجود کنترل مرکزی در دسته های مشابه در کنار یکدیگر جمع آوری کنند. با الهام گرفتن از این مدل، خوشه‌بندی مورچه‌ای توسعه داده شده است و دارای کاربردهای زیادی در علم می‌باشد. در این مدل، مورچه ها و داده هایی که باید خوشه‌بندی شوند بر روی صفحه ای دو بعدی قرار داده می‌شوند. مورچه ها می‌توانند داده هایی را که به داده های اطراف خود شبیه نیستند از جای خود در صفحه بر داشته و در کنار داده های مشابه قرار دهند. مورچه ها برای ارزیابی اینکه داده ای به داده ی اطراف خود شبیه است یا خیر دارای دید محدودی می‌باشند. این مورچه ها، همانند مورچه های واقعی از کنترل مرکزی برخوردار نیستند و به صورت مستقل از یکدیگر عمل می‌کنند. . در هر سلول شبکه همزمان دو داده یا دو مورچه نمی‌توانند قرار بگیرند و چنانچه مورچه‌ای از یک طرف شبکه بیرون رود از طرف دیگر وارد خواهد شد. مورچه ها با احتمالات زیر داده $i$ را از جای خود برداشته یا در مکانی قرار می‌دهند [14].

$$P_{pick}(i) = \left(\frac{k_{pick}}{k_{pick} + f(i)}\right) \quad (۱)$$

$$P_{drop}(i) = \left(\frac{f(i)}{k_{drop} + f(i)}\right) \quad (۲)$$

$k_{drop}$ و $k_{pick}$ پارامترهای الگوریتم می‌باشند و $f(i)$ تابعی است که شایستگی داده $i$ را برای بودن در مکانی که قرار دارد محاسبه می‌کند. چنانچه داده ی $i$ شبیه داده هایی که در همسایگی اش هستند باشد، $f(i)$ مقادیر بالا خواهد داشت و در نتیجه داده $i$ با احتمال کمی‌برداشته می‌شود (فرمول (۱)) و با احتمال بالایی در آن مکان انداخته می‌شود(فرمول (۲)). تفاوت داده ی $i$ با داده های همسایه اش باعث می‌شود که $f(i)$ مقادیر پایین داشته باشد و در این حالت احتمال اینکه داده توسط مورچه‌ای از جای خود برداشته شود زیاد می‌شود و احتمال گذاشته شدن در آن مکان برای داده پایین می‌آید. تابع ارائه شده در [3] امکان مقایسه ی داده های عددی را فراهم نمی‌کند ولی این مشکل در [14] حل و تابع زیر ارائه شد.

$$f(i) = \max\left(0, \frac{1}{\sigma^2} \sum_{j \in Neigh(i)} \left(1 - \frac{d(i,j)}{\alpha}\right)\right) \quad (۳)$$

$d(i,j)$ در محدوده ی [0,1] قرار دارد و تفاوت داده های $i$ و $j$ را تعیین می‌کند. $\alpha$ پارامتر مقیاس[۱۴] است که در محدوده ی [0,1] قرار دارد و باید با توجه به داده ها تعیین شود. $\sigma$ محدوده دید را تعیین می‌کند به این معنی که محدوده ی دید یک مورچه با پارامتر $\sigma$ در هر۴ جهت برابر است با $\frac{(\sigma-1)}{2}$. $Neigh(i)$ مجموعه ی داده هایی است که در همسایگی $\sigma$ داده ی $i$ بر روی صفحه قرار دارند.. در [5] پارامتر جدیدی به نام $step_size$ به الگوریتم اضافه شد. در این مدل مورچه ها به جای حرکت از یک سلول به سلول مجاور می‌توانند به سلولهایی که مجموع فاصله ی عمودی و افقی آنها حد اکثر $step_size$ باشد پرش کند. همچنین در این مقاله پیشنهاد شده است که ابعاد صفحه برابر با $\sqrt{10N_{data}} \times \sqrt{10N_{data}}$ و $step_size$ برابر با $2\sqrt{10N_{data}}$ قرار داده شود. شبه کد الگوریتمی را که در این مقاله مورد بررسی قرار گرفته و برگرفته از [5] می‌باشد می‌توانید در شکل ۱ ببینید. در این مدل هر مورچه برای برداشتن داده ها به لیستی که از داده های آزاد (داده هایی که توسط دیگر مورچه ها حمل نمی‌شوند) وجود دارد مراجعه می‌کند و یکی از داده ها را به صورت اتفاقی انتخاب می‌کند و آن را با احتمالی که فرمول (۱) تعیین می‌کند بر می‌دارد. این کار تا زمانی ادامه پیدا می‌کند که مورچه موفق به برداشتن یک داده بشود. پس از آن مورچه در تکرار های بعدی سعی خواهد کرد که داده اش را در مکان مناسبی قرار دهد. در هنگام انداختن داده مورچه اقدام به جهش های اتفاقی با حداکثراندازه ی $step_size$ می‌کند و از فرمول (۲) برای تصمیم گیری در مورد انداختن داده تصمیم می‌گیرد. چنانچه مورچه تصمیم به انداختن داده ای کرد و سلول مورد نظر توسط داده ی دیگری اشغال شده بود، مورچه داده را به صورت اتفاقی در یکی از سلول های همسایه می‌اندازد و چنانچه تمام سلولهای همسایه داده در خود داشتند مورچه از انداختن داده صرف نظر می‌کند.

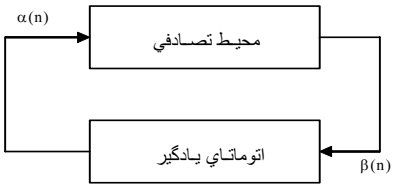
```
Algorithm Ant Clustering
INITIALIZATION PHASE
Randomly scatter data items on the toroidal grid
 for each j in 1 to #agents do
    i := random select(remaining items)
    pick up(agent(j), i)
    g := random select(remaining empty grid locations)
    place agent(agent(j), g)
 end for
 for each it_ ctr in 1 to #iterations do
    j := random select(all agents)
    step(agent(j), stepsize)
    i := carried item(agent(j))
    drop := drop item?(f(i)) // see equation 2
    if drop = TRUE then
        while pick = FALSE do
            i := random select(free data items)
            pick := pick item? // see equation 1
        end while
    end if
end for
```

**شکل ۱: الگوریتم خوشه‌بندی مورچه‌ای**

## ۳. اتوماتای سلولی، یادگیر، سلولی یادگیر

برای آشنایی با مدل CLA-PSO ابتدا باید با مدلهای اتوماتای سلولی[۱۵]، یادگیر[۱۶] و اتوماتای یادگیر سلولی[۱۷] آشنا بود. **اتوماتای یادگیر**[17] یک مدل انتزاعی است که تعداد معدودی عمل را می‌تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده و حالت درونی خود را به روز می‌کند و دوباره عمل خود را برای مرحله بعد انتخاب می‌کند. اتوماتا با تعامل با محیط عمل بهینه را فراگیری می‌کند و به این شکل پاداش دریافتی خود را از محیط حد اکثر می‌کند. شکل ۲ ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد

.




**شکل ۲: ارتباط بین اتوماتای یادگیر و محیط**

محیط را می‌توان توسط یک سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه $\alpha$ ها می‌باشد. هر گاه $\beta$ مجموعه دو عضوی باشد ، محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. اتوماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می‌گردد [17]. اتوماتای یادگیر با ساختار متغیر[۱۸] توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه ورودیهای اتوماتا و $p = \{p_1, p_2, ... p_r\}$ بردار احتمال انتخاب هر یک از اعمال و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاها، اگر عمل $\alpha_i$ در مرحله n ام انتخاب شود و این عمل، پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می‌یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می‌یابند. این تغییرات به گونه ای صورت می‌پذیرد که جمع احتمالات برابر با یک باقی بماند.. فرمولهای (۴) و (۵) یکی از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر را نشان می‌دهد.

الف- پاسخ مطلوب برای عمل i:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)\,p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \qquad (۴)$$

ب- پاسخ نامطلوب برای عمل i:

$$\begin{aligned} p_i(n+1) &= p_i(n) - (1-b)\,p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)\,p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \qquad (۵)$$

$a$ و $b$ به ترتیب پارامتر پاداش و پارامتر جریمه می‌باشد. زمانی که a و b با هم برابر باشند، الگوریتم $L_{R-P}$ [۱۹] ، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم $L_{R-\varepsilon P}$ [۲۰] و زمانی که $b$ مساوی صفر باشد، الگوریتم $L_{R-I}$ [۲۱] نامیده می‌شود.

**اتوماتای سلولی**[3][18] از یک آرایه از سلولها با تعداد ابعاد محدود تشکیل شده است. هر سلول مجموعه ای محدود از حالات را می‌تواند اختیار کند. حالت بعدی هر سلول تابعی از حالت خود و همسایه هایش در حال حاضر می‌باشد. ساده ترین نوع اتوماتای سلولی یک بعدی می‌باشد که هر سلول آن حالت صفر و یا یک را می‌توانند انتخاب کند.

**اتوماتای یادگیر سلولی** ترکیبی از اتوماتای سلولی و اتوماتای یادگیر می‌باشد. در اتوماتای یادگیر سلولی [2] [15] پاداش و جریمه هر اتوماتا بستگی به حالت خود و همسایه هایش دارد. اتوماتای سلولی در [15] به صورت رسمی به این شکل تعریف شده است :

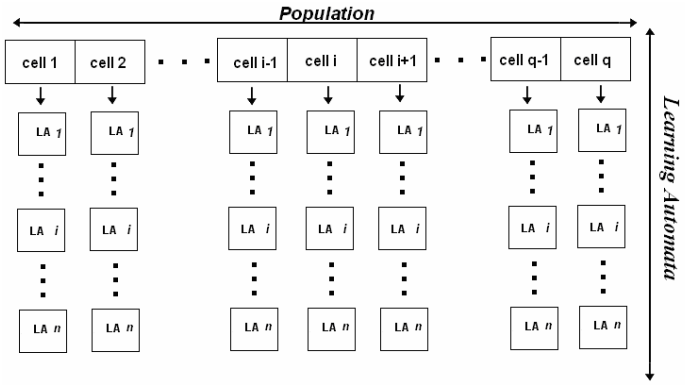
اتوماتای یادگیر سلولی d بعدی یک چندتایی $CLA = (Z^d, \phi, A, N, F)$ است به طوریکه:

- $Z^d$ یک شبکه از d تایی های مرتب از اعداد صحیح می‌باشد. این شبکه می‌تواند یک شبکه متناهی، نیمه متناهی یا متناهی باشد.
- $\phi$ یک مجموعه متناهی از حالتها می‌باشد که هر سلول می‌تواند در آن حالت قرار بگیرد..
- A، یک مجموعه از اتوماتاهای یادگیر (LA) است که هر یک از آنها به یک سلول از اتوماتای سلولی نسبت داده می‌شود.
- $N = \{\overline{x}_1, \ldots, \overline{x}_{\overline{m}}\}$ یک زیر مجموعه متناهی از $Z^d$ می‌باشد که بردار همسایگی نامیده می‌شود.
- $F : \underline{\phi}^{\overline{m}} \rightarrow \underline{\beta}$ قانون محلی CLA می‌باشد به طوریکه $\underline{\beta}$ مجموعه مقادیری است که می‌تواند به عنوان سیگنال تقویتی پذیرفته شود.

## ۴. PSO گسسته بر پایه ی اتوماتای سلولی یادگیر

PSO از مجموعه ای از ذرها تشکیل شده که در فضای مسئله در حال پرواز هستند هر فرد در جمعیت دارای حافظه ای می‌باشد که بهترین مکانی را که آن فرد تا کنون در آنجا بوده ذخیره می‌کند و *PB(Personal Best)* نام دارد. افراد جمعیت بعد از هر جابجایی *PB* خود را در صورت نیاز به روز می‌کنند و برای حرکت بعدی به سمت *PB* خود و بهترین *PB* جمعیت *GB(Global Best)* سرعت می‌گیرند. در مدل CLA-PSO هر فرد یک سلول می‌باشد که به تعداد ابعاد دودیی مسئله دارای اتوماتای یادگیر می‌باشد. هر اتوماتای یادگیر در هر سلول متناظر با یکی از ابعاد دودویی مسئله می‌باشد و وظیفه ی یادگیری احتمال صفر یا یک بودن آن بیت از جواب را بر عهده دارد. حالت هر سلول از کنار گذاشتن انتخاب های اتوماتاهایش تعیین می‌شود. با توجه به اینکه در این مدل اتوماتاها دارای دو عمل می‌باشند، هر سلول با توجه به حالتش در یک رأس یک ابر مکعب قرار می‌گیرد. شکل ۳ چینش اتوماتای یادگیر را در یک اتوماتای سلولی یک بعدی که در CLA-PSO به کار رفته نشان می‌دهد.

[3] Cellular Automata




**شکل ۳: چینش اتوماتاهای یادگیر در سلولها در توپولوژی یک بعدی**

هر سلول دارای تعدای همسایه می‌باشد و این همسایگی متقارن می‌باشد. همسایه های هر سلول در CLA-PSO برابر با سلول سمت چپ و راست آن می‌باشد. در این مدل، فرد $i$ جمعیت بهترین جوابی را که تا کنون یافته در متغیری به نام $PB_i$ ذخیره می‌کند. در طول اجرا، هر اتوماتای یادگیر در هر سلول با توجه به بردار احتمالات خود عملی (۰ یا ۱) را انتخاب می‌کند. از کنار گذاشتن این انتخابها، حالت هر سلول(ذره) جمعیت تعیین می‌شود. چنانچه این نقطه ی جدید در فضای مسئله دارای شایستگی[۲۲] بیشتری باشد، $PB$ به روز می‌شود. بعد از انتخاب عمل توسط اتوماتای یادگیر، چنانچه این انتخاب با بیت متناظر در $PB$ و $LB$ (که بهترین $PB$ در بین همسایه های آن سلول می‌باشد) برابر باشد، اتوماتای یادگیر پاداش می‌بینید و در غیر این صورت جریمه می‌شود. بعد از هر یادگیری احتمال یک بودن و صفر بودن هر اتوماتای یادگیر با متغیری به نام $P_{max}$ مقایسه می‌شود. چنانچه یکی از این احتمالات از این پارامتر بزرگتر بود برابر $P_{max}$ قرار داده می‌شود و دیگری طوری تغییر می‌کند که جمع احتمالات برابر با یک باقی بماند. شکل ۴ شبه کد این الگوریتم را نشان می‌دهد.

```
While not done do
    For each particle i in CLA do
        Generate a new corner of hypercube and go to that location
        Evaluate the new corner of hypercube
        If fitness(new location)> fitness(PB_i) then
            Update PB_i
        End if
        Select LB_i cells from neighbors of cell i
        Generate the reinforcement signal vector
        Update LAs of particle i
        Correct out bound probabilities of LA in particle i
    End for
End while
```

**شکل ۴: شبه کد الگوریتم CLA-PSO**

## ۵. تنظیم پارامترهای خوشه‌بندی مورچه‌ای به کمک CLA-PSO

همانطور که دیدیم، خوشه‌بندی مورچه‌ای دارای پارامترهای زیادی می‌باشد که تنظیم آنها برای کارایی بالا بسیار دشوار می‌باشد. بعضی از این پارامترها عبارتند از $K_{drop}$، $K_{pick}$، $step_size$، $\alpha$ و $\sigma$. در این مقاله از CLA-PSO برای تطبیق این پارامترها در خوشه‌بندی مورچه‌ای استفاده کرده ایم. ابعاد صفحه ای که خوشه‌بندی بر روی آن انجام می‌شود، طبق توصیه [5] برابر $\sqrt{10\,N_{data}} \times \sqrt{10\,N_{data}}$ قرار داده شده است.

در این مقاله از دو نوع مورچه برای خوشه بندی استفاده کرده ایم که هرکدام نیمی‌از زمان اجرای خوشه‌بندی را در اختیار دارند و از CLA-PSO برای تطبیق پارامترهای این دو نوع مورچه استفاده کرده‌ایم. برای هر نوع از این مورچه‌ها ۵ پارامتر $K_{drop}$، $K_{pick}$، $step_size$، $\alpha$ و $\sigma$ را تنظیم کرده ایم، بنابراین ۱۰ پارامتر برای این دو نوع مورچه باید تنظیم شوند. ذره های CLA-PSO

رشته‌های دودویی هستند که نشان دهنده یک ترکیب از این ۱۰ پارامتر برای تنظیم می‌باشد. جدول ۲ نشان دهنده ی تعداد بیتهای استفاده شده برای کد کردن پارامتر به شکل دودویی و حدود آنهاست.

**جدول ۲: تعداد بیتهای مورد استفاده برای کد کردن و حدود پارامترها**

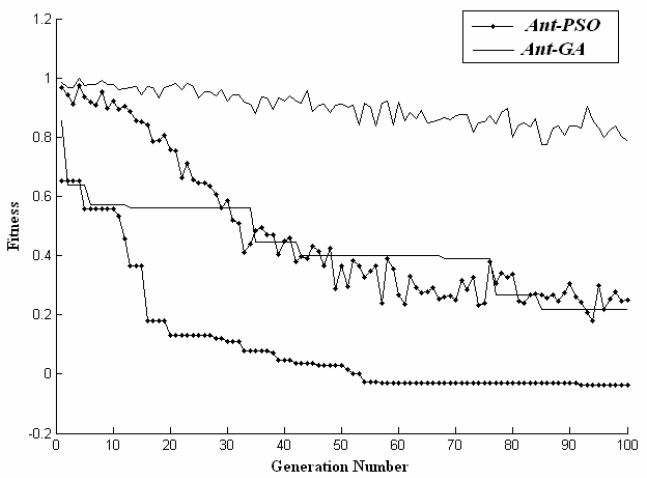
| | $\alpha$ | $\sigma$ | $step_size$ | $K_{pick}$ | $K_{drop}$ |
|---|---|---|---|---|---|
| حدود | [0,1] | $2i+1$ ($i$=1...8) | [1…128] | [0,1] | [0,1] |
| تعداد بیتها | 4 | 3 | 4 | 4 | 4 |

برای ارزیابی شایستگی هر ذره، رشته باینری آن ذره به ۱۰ پارامتر دهدهی تبدیل شده و از آن پارامترها برای عمل خوشه‌بندی استفاده می‌شود. شایستگی هر ذره برابر با کیفیت خوشه‌بندی در نظر گرفته می‌شود. بعد از ارزیابی هر ذره مراحلCLA-PSO اجرا شده و نسل به نسل ذره‌ها (پارامترهای خوشه‌بندی) بهبود داده می‌شوند. تعداد جمعیت را برابر با ۳۰ و تعداد نسل‌ها را برابر با ۱۰۰ قرار داده‌ایم. برای تنظیم توسط الگوریتم ژنتیک، به شکل مشابه عمل شده با این تفاوت که از یک نوع مورچه برای خوشه‌بندی استفاده شده است[1]. بنابر این افراد جمعیت الگوریتم ژنتیک نشان دهنده ی ۵ پارامتر می‌باشند. برای ارزیابی خوشه‌بندی ذره ها در هنگام اجرای CLA-PSO و افراد در الگوریتم ژنتیک از مجموعه داده ای ۴ کلاسه با توزیع نرمال درای میانگین های (0.2,0.2) و (0.2,0.8) و (0.8,0.2) و (0.8,0.8) با انحراف معیار 0.05 استفاده کردیم. هر کلاس دارای ۱۰۰ داده می‌باشد. کیفیت خوشه‌بندی با استفاده از فرمول زیر تعیین می‌شود.

$$f_i = \sum_{\sigma\in\Sigma}[\sum_{j\in Neigh(i,\sigma)}[-c(i,j)+2(1-c(i,j))]]+(\sigma^2-|Neigh(i,\sigma)|)$$
$$f = \sum_{i=1}^{N_{data}} f_i / N_{data} \qquad (۶)$$

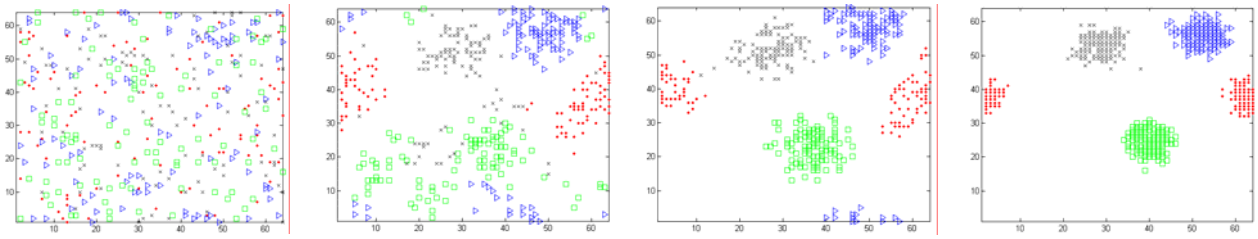
در فرمول (۶)، چنانچه داده های $i$ و $j$ در یک کلاس باشند $c(i,j)$ برابر با یک است و در غیر اینصورت برابر با ۰ می‌باشد. $Neigh(i,\sigma)$ نشان دهنده ی مجموعه داده های همسایه های داده $i$ در همسایگی $\sigma$ بر روی صفحه خوشه‌بندی می‌باشد و $|\ |$ تعداد اعضا یک مجموعه را نشان می‌دهد و $\Sigma=\{3,5,7,9,11\}$. برای محاسبه ی $f_i$، کلاس داده ی $i$ با کلاسهای داده های همسایه مقایسه می‌شود. چنانچه کلاسها برابر نباشند عدد ۲+ $(2(1-c(i,j)))$و چنانچه برابر باشد عدد ۱-$(-c(i,j))$ و فضای خالی عدد ۱+ $(\sigma^2-|Neigh(i,\sigma)|)$در نظر گرفته می‌شود و با $f_i$جمع می‌شود. تابع $f$ باید حداقل شود. به این شکل، بودن داده ای از یک کلاس متفاوت در کنار داده ی $i$ یا خالی بودن همسایگی از داده باعث افزایش این معیار می‌شود. تاثیر داده با کلاس متفاوت دو برابر تاثیر فضای خالی است. به بیان بهتر، جریمه ی داده ای از کلاس متفاوت دو برابر فضای خالی است. همچنین، بودن داده ای از کلاس یکسان باعث کاهش این معیار می‌باشد. این کاهش برابر با افزایش فضای خالی و نصف افزایش داده با کلاس متفاوت می‌باشد. $f_i$ برای مقادیر مختلف $\sigma$ محاسبه می‌شود. دلیل اینست که $\sigma$ های کوچک، چگالی محلی را ارزیابی می‌کنند و $\sigma$ های بزرگ جدایی خوشه ها را برروی صفحه ارزیابی کند و ترکیب این دو می‌تواند به خوبی کیفیت خوشه‌بندی را ارزیابی کند.

همانطور که از جدول ۲ مشخص است هر فرد در الگوریتم ژنتیک دارای ۱۹ بیت و در CLA-PSO دارای ۳۸(=۱۹×۲) بیت می‌باشد. CLA مورد استفاده در CLA-PSO یک بعدی می‌باشد و اتوماتای یادگیر از قانون یادگیری $L_{RP}\ (a=b=0.1)$ استفاده می‌کند و $P_{max}$ برابر با ۰/۹۵ قرار داده شده است. الگوریتم ژنتیک استفاده شده در [1] به این شکل می‌باشد: ٪۱۰ از بهترین افراد در هر نسل بدون تغییر به نسل بعد منتقل می‌شوند. برای بازترکیبی ۲ نفر از ٪۱۰ برتر به شکل اتفاقی انتخاب شده و یک فرزند تولید می‌شود. هر کدام از بیتهای فرزند به صورت اتفاقی از یکی از والدین انتخاب می‌شود. برای جهش نیز هر بیت با احتمال ۰/۰۱ تغییر می‌کند. شکل ۵ میانگین و بهترین شایستگی را در بین افراد جمعیت CLA-PSO و الگوریتم ژنتیک را نشان می‌دهد. در هر دو روش تعداد مورچه برابر با ۱۰ قرار داده شده اند.



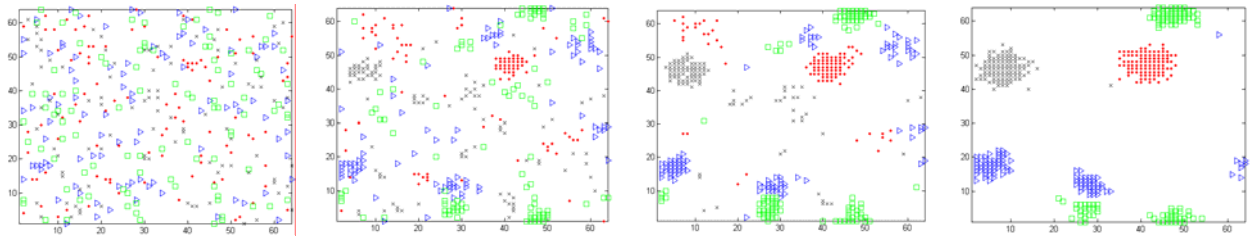

**شکل ۵: میانگین و بهترین شایستگی در بین افراد جمعیت CLA-PSO و الگوریتم ژنتیک**

همانطور که می‌بینید، میانگین شایستگی و بهترین شایستگی، با استفاده از CLA-PSO بهتر از الگوریتم ژنتیک می‌باشد. شکلهای ۶ و ۷ مراحل خوشه‌بندی توسط پارامترهای بدست آمده توسط CLA-PSO (Ant-PSO) و الگوریتم ژنتیک (Ant-GA) را نشان می‌دهد.



الف ب پ ت

**شکل ۶: مراحل خوشه‌بندی Ant-PSO**



الف ب پ ت

**شکل ۷: مراحل خوشه‌بندی Ant-GA**

همانطور که شکلهای ۶ و ۷ نشان می‌دهند، شیوه ی خوشه‌بندی پارامترهای بدست آمده توسط CLA-PSO با پارامترهای الگوریتم ژنتیک متفاوت می‌باشد. در Ant-PSO مورچه های نوع اول خوشه‌بندی خام تولید می‌کنند(شکل ۶ الف،ب) و پس از آن مورچه های نوع دوم خوشه های خام تشکیل شده توسط مورچه های نوع اول را پالایش می‌کنند(شکل ۶ پ،ت). مقایسه ی شکلهای ۶ و ۷ نشان می‌دهد که پارامترهای بدست آمده توسط CLA-PSO بهتر از الگوریتم ژنتیک عمل می‌کنند. پارامترهای بدست آمده توسط CLA-PSO و الگوریتم ژنتیک را می‌توانید در جدول ۳ ببینید.

**جدول ۳: پارامترهای بدست آمده توسط CLA-PSO و الگوریتم ژنتیک**

| Parameter | *Ant1-PSO* | *Ant2-PSO* | *Ant-GA* |
|---|---|---|---|
| $K_{pick}$ | 0.025 | 0.062 | 0.062 |
| $K_{drop}$ | 0.062 | 0.625 | 0.187 |
| $\alpha$ | 0.46 | 0.375 | 0.375 |
| *Sigma* | 13 | 7 | 7 |
| *Step size* | 128 | 128 | 128 |

## ۶. ارزیابی

برای ارزیابی دقیقتر پارامترهای بدست آمده توسط CLA-PSO و الگوریتم ژنتیک باید آنها را برای خوشه‌بندی داده های دیگری نیز به کار برد. برای این منظور باید تغییراتی در پارامترهای بدست آمده توسط CLA-PSO و الگوریتم ژنتیک بدهیم تا قابل اعمال به تمام داده ها باشد. مشکلی که در این زمینه وجود دارد اینست که کارایی الگوریتم بستگی زیادی به پارامتر $\alpha$ دارد و پارامتر $\alpha$ با توجه به نوع داده ها مقادیر متفاوتی باید داشته باشد. برای حل این مشکل از پارامتر $\alpha$ که توسط CLA-PSO و الگوریتم ژنتیک بدست آمده صرف نظر می‌کنیم و آن را به صورت آزمایش و خطا بدست می‌آوریم. برای تعیین $\alpha$ مورچه ی دوم در Ant-PSO آن را $(\frac{\alpha_{Ant2}}{\alpha_{Ant1}} = \frac{0.375}{0.46} =)0.8151$ برابر $\alpha$ مورچه ی اول قرار می‌دهیم. *step_size* نیز نقش مهمی در کارایی الگوریتم دارد. در آزمایش بخش ۴ دیدیم که *step_size* بهینه در تمام موارد برابر با 128 که برابر با $2\sqrt{10N_{data}}$ است بدست آمد. به همین دلیل برای اعمال پارامتر ها به داده های مختلف *step_size* را برای مورچه ها در تمام آمایش ها برابر با $2\sqrt{10N_{data}}$ قرار داده ایم.

برای ارزیابی خوشه‌بندی انجام شده توسط روش ارائه شده (Ant-PSO) و روش موجود(Ant-GA) و K-means از ۴ معیار ارزیابی برای خوشه‌بندی ۵ داده استفاده کرده ایم. ۴ معیار مورد استفاده عبارتند از :

**۱.F-Measure** : این معیار دو مفهوم دقت[۲۳] و به یاد آوری[۲۴] که از مفاهیم بازیابی اطلاعات هستند، استفاده می‌کند. این معیارو مقداری بین ۰ و ۱ دارد ک مقدایر نزدیک به ۱ این معیار مطلوب هستند.

$$p(i,j) = \frac{n_{ij}}{n_j} \qquad r(i,j) = \frac{n_{ij}}{n_i}$$
$$F(i,j) = \frac{(b^2+1).\,p(i,j).r(i,j)}{b^2.\,p(i,j) + r(i,j)} \qquad (۷)$$
$$F = \sum_i \frac{n_i}{n} \max_j \{F(i,j)\}$$

در فرمولهای فوق، $n_i$ و $n_j$ و $n_{ij}$ به ترتیب برابراند با تعداد داده های کلاس $i$ و تعداد داده های خوشه ی $j$ (بدست آمده توسط الگوریتم خوشه‌بندی) و تعداد داده های کلاس $i$ در خوشه ی $j$. $p$ و $r$ نیز به ترتیب نشان دهنده ی مفاهیم دقت و یادآوری می‌باشند. در این مقاله از $b=1$ استفاده شده است.

**۲.Rand Index** : فرض کنید که V آرایه ای باشد که خوشه ی داده ها (بدست آمده توسط الگوریتم خوشه‌بندی) و U آرایه باشد که کلاس داده ها را نشان می‌دهد. این معیار به این شکل تعریف می‌شود و مقداری بین ۰ و ۱ دارد. مقدایر نزدیک به ۱ این معیار مطلوب هستند.

$$\begin{aligned} a &= \{i,j \mid U(i) = U(j) \wedge V(i) = V(j)\} \\ b &= \{i,j \mid U(i) = U(j) \wedge V(i) \neq V(j)\} \\ c &= \{i,j \mid U(i) \neq U(j) \wedge V(i) = V(j)\} \\ d &= \{i,j \mid U(i) \neq U(j) \wedge V(i) \neq V(j)\} \end{aligned} \qquad (۸)$$
$$R = \frac{a+d}{a+b+c+d}$$

**۳.Entropy** : این پارامتر اعدادی بین ۰ و ۱ تولید می‌کند و اعداد نزدیک به صفر مطلوب می‌باشند.

$$p_{ij} = I(i,j)/n_i$$
$$e_i = -\sum_j p_{ij} \log p_{ij} \qquad (۹)$$
$$E = \sum_i e_i n_i / n$$

که در آن $I(i,j)$ تعداد داده های مشترک بین خوشه ی $i$ و کلاس $j$ می‌باشد و $n_i$ و $n$ به ترتیب تعداد داده های خوشه ی $i$ و تعداد کل داده ها می‌باشد.

**۴.واریانس درون کلاسی(Variance) :**

$$V = \sum_{c \in C} \sum_{i \in c} d(\mu_c, i)^2 \qquad (۱۰)$$

$d$ و $C$ و $\mu_c$ به ترتیب تابع فاصله ی مورد استفاده در خوشه‌بندی، مجموعه ی خوشه ها و میانگین داده های خوشه ی $c$ می‌باشند.

۵ داده ی مورد استفاده عبارتند از: **Iris.۱** : دارای ۱۵۰ داده ی ۴ بعدی می‌باشد که در ۳ کلاس به شکل مساوی تقسیم شده اند. **Wisconsin.۲**: از ۶۸۳ داده ی ۱۰ بعدی تشکیل شده ودارای ۲ کلاس با تعداد داده های ۲۳۹ و ۴۴۴ می‌باشد. **Ecoli.۳** : از ۳۳۶ داده ی ۷ بعدی تشکیل شده و دارای ۷ کلاس با تعداد داده های ۱۴۳، ۷۹، ۵۲، ۳۵، ۲۰، ۵ و ۲ می‌باشد. **Data1.۴**: از ۱۰۰۰ داده ی ۲ بعدی تشکیل شده است که به شکل مساوی در ۴ کلاس با توزیع نرمال با میانگین های (5,2)، (-5,2)، (5,-2) و (-5,-2) و انحراف معیار2 تقسیم شده اند. **Galss.۵** که شامل ۲۱۴ داده ی ۹ بعدی است و به ۶ کلاس که دارای ۷۶، ۷۰، ۲۹، ۱۷، ۱۳ و ۹ داده می‌باشند تقسیم شده است.

تمام داده ها در تمام ابعاد قبل از استفاده نرمال شده اند. برای اینکه ماتریس $d$ (ر.ک. فرمول (۳)) اعدادی بین ۰ و ۱داشته باشد، اعداد این ماتریس نرمال شده اند. برای محاسبه ی ماتریس $d$ برای تمام داده ها به غیر از Iris از فاصله ی اقلیدسی استفاده شده است و برای داده ی Iris از فاصه ی کسینوسی استفاده شده است. پارامترهای وابسته به داده ها که در هر دو روش به صورت یکسان مورد استفاده قرار گرفته اند در جدول ۴ ذکر شده اند

**جدول ۴: پارامترهای مورد استفاده در خوشه‌بندی برای داده های مختلف**

| | $\alpha_{Ant1}$ (Ant-PSO) | $\alpha$ (Ant-GA) | تعداد تکرار ها |
|---|---|---|---|
| Iris | 0.05 | 0.04 | 90,000 |
| Wisconsin | 0.6 | 0.6 | 540,000 |
| Ecoli | 0.44 | 0.35 | 350,000 |
| Data1 | 0.5 | 0.5 | 150,000 |
| Glass | 0.33 | 0.33 | 300000 |

جدول ۵ میانگین ۴ معیار را برای ۵ داده در ۲۰ تکرار گزارش می‌کند.

**جدول ۵: مقایسه ی خوشه‌بندی روشهای Ant-PSO و Ant-GA و Kmeans برای ۵ داده و ۴ معیار**

| **Wisconsin** | ***k-means*** | ***Ant-GA*** | ***Ant-PSO*** |
|---|---|---|---|
| F-Measure | 0.9699 | 0.6328 | 0.9557 |
| Rand Index | 0.9240 | 0.5278 | 0.8870 |
| Entropy | 0.1054 | 0.4021 | 0.1050 |
| Variance | 166.0238 | 260.7895 | 171.0297 |
| **IRIS** | ***k-means*** | ***Ant-GA*** | ***Ant-PSO*** |
| F-Measure | 0.8478 | 0.8985 | 0.9472 |
| Rand Index | 0.8627 | 0.8807 | 0.9313 |
| Entropy | 0.2519 | 0.3207 | 0.2007 |
| Variance | 16.4638 | 16.7332 | 16.3139 |
| **Data1** | ***k-means*** | ***Ant-GA*** | ***Ant-PSO*** |
| F-Measure | 0.9488 | 0.8774 | 0.9830 |
| Rand Index | 0.9526 | 0.9026 | 0.9834 |

| | | | |
|---|---|---|---|
| Entropy | 0.1787 | 0.3319 | 0.0849 |
| Variance | 55.2279 | 74.0501 | 51.5061 |
| **Ecoli** | ***k-means*** | ***Ant-GA*** | ***Ant-PSO*** |
| F-Measure | 0.6512 | 0.6749 | 0.7625 |
| Rand Index | 0.7991 | 0.7954 | 0.8696 |
| Entropy | 0.5156 | 0.6359 | 0.6057 |
| Variance | 38.0333 | 44.9740 | 43.6553 |
| **Glass** | ***k-means*** | ***Ant-GA*** | ***Ant-PSO*** |
| F-Measure | 0.4333 | 0.4238 | 0.5040 |
| Rand Index | 0.6692 | 0.6321 | 0.5937 |
| Entropy | 1.0769 | 1.2043 | 1.1699 |
| Variance | 29.6713 | 34.7015 | 32.1147 |

نتایج جدول ۵ نشان می‌دهد که استفاده از دو نوع مورچه و تطبیق پارامترهای آن توسط CLA-PSO بهتر از استفاده از یک نوع مورچه و تطبیق پارامترهای آن با الگوریتم ژنتیک عمل می‌کند. جدول بالا همچنین کارایی بالای روش ارائه شده را در مقایسه با K-Means نشان می‌دهد.

## ۷. نتیجه گیری

در این مقاله از CLA-PSO برای تطبیق پارامترهای خوشه‌بندی مورچه‌ای استفاده کردیم. برای خوشه‌بندی از ۲ نوع مورچه استفاده کردیم و نتایج را با K-Means و تنها روش موجود برای تطبیق پارامترهای مورچه بندی خوشه ای که مبتنی بر الگوریتم ژنتیک است و از یک نوع مورچه استفاده می‌کند مقایسه کردیم. مقایسه ها نشان می‌دهد که روش ارائه شده پارامترهایی تولید می‌کند که کارایی بالایی در مقایسه با پارامترهای بدست آمده توسط الگوریتم ژنتیک و روش خوشه‌بندی K-means دارند.

## مراجع

[1] Aranha, C., Iba, H.; "*The Effect of Using Evolutionary Algorithms on Ant Clustering Techniques*", Proceedings of the 2006 Asia Pacific Workshop on Genetic Programming (ASPGP06), pp. 24-34, 2006.

[2] Beigy, H. and Meybodi, M. R.; "*A Mathematical Framework for Cellular Learning Automata*", Advances on Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004.

[3] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chr´etien, L.; "*The dynamics of collective sorting: Robot-like ants and ant-like robots*"; Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 1, pp. 356–365. MIT Press, Cambridge, MA, 1991.

[4] Goldberg, D. E.; *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989

[5] Handl, j., Knowles, J., and Dorigo, M.; "*Strategies for the increased robustness of ant-based clustering*", Lecture Notes in Computer Science, Vol. 2977, pp. 90–104, 2004.

[6] Handl, j., and Meyer, B.; "*Improved ant-based clustering and sorting in a document retrieval interface*", Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature, volume 2439 of LNCS, pp. 913–923. Springer-Verlag, Berlin, Germany, 2002.

[7] Hoe, K., Lai, W., and Tai. T.; "*Homogenous ants for web document similarity modeling and categorization*", Proceedings of the Third International Workshop on Ant Algorithms (ANTS 2002), volume 2463 of LNCS, pp. 256–261. Springer-Verlag, Berlin, Germany, 2002.

[8] Jafarpour, B., and Meybodi, M. R.; "*A Hybrid Method for Optimization (Discrete PSO + CLA)*" , Accepted in IEEE International Conference on Intelligent and Advanced Systems, Malaysia, 2007.

[9] Kuntz, P., and Snyers, D.; "*New results on an ant-based heuristic for highlighting the organization of large graphs*", Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1451–1458. IEEE Press, Piscataway, NJ, 1999.

[10] Kuntz, P., Layzell, P., and Snyers, D.; "*A colony of ant-like agents for partitioning in VLSI technology*", Proceedings of 4th European Conference on Artificial Life, MIT Press, July 1997.

[11] Kuntz, P., Snyers, D., and Layzell, P.; "*A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning*". Journal of Heuristics, pp. 327–351, 1998.

[12] Kennedy, J., and Eberhart, R. C.; "*Particle Swarm Optimization*", Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, , pp. 1942–1948, 1995.

[13] Kennedy, J., and Eberhart, R. C.; "*A Discrete Binary Version of The Particle Swarm Algorithm*", Proceedings of Conference on Systems, Man, and Cybernetics, pp. 4104-4108, IEEE Service Center, Piscataway, NJ, 1997.

[14] Lumer, E. D., and Faieta, B.; "*Diversity and adaptation in populations of clustering ants*", Proc. of the Third International Conference on The Simulation of Adaptative Behavior: From Animals to Animats 3, MIT Press, pp. 449–508, 1994.

[15] Meybodi, M. R., Beigy, H., and Taherkhani, M.; "*Cellular Learning Automata*", Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, pp. 153-163, 2001.

[16] Rastegar, R., Meybodi, M. R. and Badie, K.; "*A New Discrete Binary Particle Swarm Optimization based on Learning Automata*", Proceedings of International Conference on Machine Learning and Applications (ICMLA2004), pp.456-462, USA, IEEE Press, 2004.

[17] Thathachar, M. A. L., Sastry, P. S.; "*Varieties of Learning Automata: An Overview*", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.

[18] Wolfram, S., *Cellular Automata and Complexity*, Perseus Books Group, 1994.

---

*1 Partitioning*
*2 Genetic Algorithm*
*3 Survival of the fittest*
*4 Selection*
*5 Recombination (Crossover)*
*6 Mutation*
*7 Particle Swarm Optimization*
*8 Artificial Life (A-Life)*
*9 Swarming Theory*
*10 Evolutionary Programming*
*11 Particle*
*12 Discrete PSO based on Cellular Learning Automata*
*13 Aranha*
*14 Scaling Parameter*
*15 Cellular Automata*
*16 Learning Automata*
*17 Cellular Learning Automata (CLA)*
*18 Variable Structure Learning Automata*
*19 Linear Reward Penalty*
*20 Linear Reward Epsilon Penalty*
*21 Linear Reward Inaction*
*22 Fitness*
*23 Precision*
*24 Recall*