

# تطبیق پارامترهای خوشه بندی مورچه ای به کمک CLA-PSO

برنا جعفرپور، محمد رضا میبیدی

دانشگاه صنعتی امیر کبیر، دانشکده مهندسی کامپیوتر و فن آوری اطلاعات،

آزمایشگاه محاسبات نرم

jafarpour@cic.aut.ac.ir, mmezbodi@aut.ac.ir

## چکیده

خوشه بندی مورچه ای روشی الهام گرفته از طبیعت می باشد دارای ویژگی های جالبی می باشد 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 می باشد که تاثیر زیادی در عملکرد و همگرایی الگوریتم دارند و معمولا به صورت آزمایش و خطا تعیین می گردند. در این مقاله روشی مبتنی بر CLA-PSO<sup>1</sup> که یک مدل گسسته ی PSO می باشد برای تطبیق اتوماتیک پارامترهای خوشه بندی مورچه ای پیشنهاد می گردد. به منظور بررسی کارایی روش پیشنهادی، این روش با تنها روش موجود تطبیق پارامترها که بر اساس الگوریتمهای ژنتیکی می باشد مقایسه گردیده است. نتایج آزمایشهای انجام گرفته حاکی از کارایی بالای روش پیشنهادی در مقایسه با روش مبتنی بر الگوریتم ژنتیکی و روش خوشه بندی k-means می باشد.

کلمات کلیدی: خوشه بندی مورچه ای، اتوماتاهای یادگیر، CLA-PSO، تنظیم پارامتر

## Adaptation of Ant Clustering Parameters using CLA-PSO

B. Jafarpour, M. R. Meybodi

Amirkabir University of Technology, Computer Engineering and IT Department,

Soft Computing Laboratory

jafarpour@cic.aut.ac.ir, mmezbodi@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 from 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 assess 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

## ۱-مقدمه

روش خوشه بندی مورچه ای که الهام گرفته از رفتار مورچه ها در طبیعت می باشد اولین بار در [1] برای مجموعه ای از ربات ها که اشیاء را دسته بندی می کردند ارائه شد. این الگوریتم در [2] برای خوشه بندی داده ها تصحیح و ارائه شد. در روش خوشه بندی مورچه ای از تعدادی مورچه با عملکرد ساده و بدون کنترل مرکزی استفاده می شود. در این روش یک شبکه ۲ بعدی از

<sup>1</sup> Discrete PSO based on Cellular Learning Automata

سلولها وجود دارد که داده‌ها و مورچه‌ها بر روی سلولهای آن قرار دارند و مورچه‌ها می‌توانند بر روی آنها حرکت کنند. در این روش بر خلاف بسیاری از روش‌های خوشه‌بندی احتیاج به از پیش تعیین کردن تعداد خوشه‌ها نمیباشد و تعداد آنها بطور خودکار در حین فرایند خوشه‌بندی تعیین میگردد. یکی دیگر از ویژگی‌های 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 [4][5]، قسمت‌بندی<sup>۲</sup> گراف [6][7] و قسمت‌بندی در VLSI [8] و غیره می‌باشد.

الگوریتم ژنتیک<sup>۳</sup> [18] روشی الهام گرفته از طبیعت برای جستجو در مسائل با فضای حالت بزرگ می‌باشد و بر پایه ی قانون بقای شایسته‌ترین<sup>۴</sup> داروین عمل می‌کند. در این روش مجموعه‌ای 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<sup>۵</sup>، بازترکیبی<sup>۶</sup> و جهش<sup>۷</sup>. عملگر 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 افراد جمعیت می‌شود. جهش در الگوریتم ژنتیک مانع از همگرایی زودرس الگوریتم شده و باعث تولید جوابهای بهتر می‌شود.

PSO<sup>۸</sup> روشی است که اولین بار برای بهینه‌سازی غیر خطی توابع پیوسته ارائه شده است [9]. این روش با کمک شبیه‌سازی مدل‌های اجتماعی بدست آمده است. این الگوریتم ریشه در دو مدل کلی دارد: ۱. حیات مصنوعی<sup>۹</sup> مانند مدل‌های گروه پرندگان و دسته ماهی‌ها و تئوری آشوب<sup>۱۰</sup>. ۲. مدل‌های تکاملی به ویژه الگوریتم ژنتیک و برنامه ریزی تکاملی<sup>۱۱</sup>. PSO شباهت زیادی به دیگر مدل‌های تکاملی مانند الگوریتم ژنتیک دارد به عنوان مثال این روش با مجموعه‌ای 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 جواب‌ها، که ذره<sup>۱۲</sup> 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 یافته‌اند 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 [9] گسسته [10][11][12]. در مدل پیوسته PSO در یک فضای حالت پیوسته جستجو می‌کند ولی در حالت گسسته فضای حالت گسسته می‌باشد. در [12] یک مدل گسسته ی PSO ارائه شده است که برخلاف سایر مدل‌های گسسته ی PSO از همگرایی زودرس رنج نمی‌برد و کارایی بالایی در بهینه‌سازی دارد.

خوشه‌بندی مورچه‌ای دارای پارامترهای متعددی از جمله پارامترهای مربوط به برداشتن، گذاشتن داده‌ها 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 [12]، که یک مدل گسسته ی دودویی PSO می‌باشد، برای تطبیق اتوماتیک پارامترهای خوشه‌بندی مورچه‌ای پیشنهاد می‌گردد. در این مقاله از دو نوع مورچه برای خوشه‌بندی استفاده کرده ایم که هر کدام 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<sup>۱۳</sup> و همکاران ارائه گردیده است [13]. مقایسه‌ها، کارایی بالای روش پیشنهادی را در مقایسه با روش موجود و K-means نشان می‌دهد.

<sup>2</sup> Partitioning

<sup>3</sup> Genetic Algorithm

<sup>4</sup> Survival of the Fittest

<sup>5</sup> Selection

<sup>6</sup> Recombination (Crossover)

<sup>7</sup> Mutation

<sup>8</sup> Particle Swarm Optimization

<sup>9</sup> Artificial Life (A-Life)

<sup>10</sup> Swarming Theory

<sup>11</sup> Evolutionary Programming

<sup>12</sup> Particle

<sup>13</sup> Aranha

## ۲- خوشه بندی مورچه ای

روش خوشه بندی مورچه ای الهام گرفته از رفتار مورچه ها در طبیعت می باشد. مورچه ها در طبیعت می توانند اشیاء مختلف را بدون وجود کنترل مرکزی در دسته های مشابه در کنار یکدیگر جمع آوری کنند. با الهام گرفتن از این مدل، خوشه بندی مورچه ای توسعه داده شده است و دارای کاربردهای زیادی در علم می باشد. در این مدل، مورچه ها و داده هایی که باید خوشه بندی شوند بر روی صفحه ای دو بعدی قرار داده می شوند. مورچه ها می توانند 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 می باشند. این مورچه ها، همانند مورچه های واقعی از کنترل مرکزی برخوردار نیستند و به صورت مستقل از یکدیگر عمل می کنند. در هر سلول شبکه همزمان دو داده یا دو مورچه نمی توانند قرار بگیرند و چنانچه مورچه ای 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 می دهند [2].

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

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

$k_{pick}$  و  $k_{drop}$  پارامترهای الگوریتم می باشند و  $f(i)$  تابعی است که شایستگی داده  $i$  را برای بودن در مکانی که قرار دارد محاسبه می کند. 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 کمی برداشته می شود (فرمول (۱)) و با احتمال بالایی در آن مکان انداخته می شود (فرمول (۲)). تفاوت داده ی  $i$  با داده های همسایه اش باعث می شود که  $f(i)$  مقادیر پایین داشته باشد و در این حالت احتمال اینکه داده توسط مورچه ای از جای خود برداشته شود زیاد می شود و احتمال گذاشته شدن در آن مکان برای داده پایین می آید. تابع ارائه شده در [1] امکان مقایسه ی داده های عددی را فراهم نمی کند ولی این مشکل در [2] حل و تابع زیر ارائه شد.

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

$d(i,j)$  در محدوده ی  $[0,1]$  قرار دارد و تفاوت داده های  $i$  و  $j$  را تعیین می کند.  $\alpha$  پارامتر مقیاس<sup>۱۴</sup> است که در محدوده ی  $[0,1]$  قرار دارد و باید با توجه به داده ها تعیین شود.  $\sigma$  محدوده دید را تعیین می کند به این معنی که محدوده ی دید یک مورچه با پارامتر  $\sigma$  در هر ۴ 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 [3] پارامتر جدیدی به نام  $step\_size$  به الگوریتم اضافه شد. در این مدل مورچه ها به جای حرکت از یک سلول به سلول مجاور می توانند به سلولهایی که مجموع فاصله ی عمودی و افقی آنها حد اکثر  $step\_size$  باشد پرش کنند. 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 [3] می باشد می توانید در جدول 1 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 نمی شوند) وجود دارد مراجعه می کند و یکی از داده ها را به صورت اتفاقی انتخاب می کند و آن را با احتمالی که فرمول (۱) تعیین می کند بر می دارد. این کار تا زمانی ادامه پیدا می کند که مورچه موفق به برداشتن یک داده بشود. پس از آن مورچه در تکرارهای بعدی سعی خواهد کرد که داده اش را در مکان مناسبی قرار دهد. در هنگام انداختن داده مورچه اقدام به جهش های اتفاقی با حداکثر اندازه ی  $step\_size$  می کند و از فرمول (۲) برای تصمیم گیری در مورد انداختن داده تصمیم می گیرد. 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 می اندازد و چنانچه تمام سلولهای همسایه داده در خود داشتند مورچه از انداختن داده صرف نظر می کند.

```

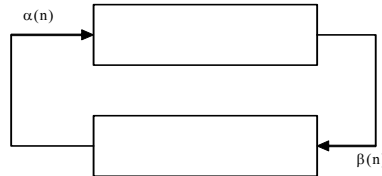
Algorithm Ant Clustering
1: INITIALIZATION PHASE
2: Randomly scatter data items on the toroidal grid
3: for each  $j$  in 1 to #agents do
4:    $i := \text{random select}(\text{remaining items})$ 
5:    $\text{pick up}(\text{agent}(j), i)$ 
6:    $g := \text{random select}(\text{remaining empty grid locations})$ 
7:    $\text{place agent}(\text{agent}(j), g)$ 
8: end for
9: for each  $it\_ctr$  in 1 to #iterations do
10:   $j := \text{random select}(\text{all agents})$ 
11:   $\text{step}(\text{agent}(j), \text{stepsize})$ 
12:   $i := \text{carried item}(\text{agent}(j))$ 
13:   $\text{drop} := \text{drop item?}(f(i))$  // see equation 2
14:  if  $\text{drop} = \text{TRUE}$  then
15:    while  $\text{pick} = \text{FALSE}$  do
16:       $i := \text{random select}(\text{free data items})$ 
17:       $\text{pick} := \text{pick item?}$  // see equation 1
18:    end while
19:  end if
20: end for

```

جدول 1 الگوریتم خوشه بندی مورچه ای

### ۳- اتوماتای سلولی، یادگیر، سلولی یادگیر

برای آشنایی با مدل CLA-PSO ابتدا باید با مدل‌های اتوماتای سلولی<sup>۱۵</sup>، یادگیر<sup>۱۶</sup> و اتوماتای یادگیر سلولی<sup>۱۷</sup> آشنا بود. اتوماتای یادگیر [14] یک مدل انتزاعی است که تعداد محدودی 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 1 ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.



شکل 1 ارتباط بین اتوماتای یادگیر و محیط

محیط را می‌توان توسط یک سه تایی  $E \equiv \{\alpha, \beta, c\}$  نشان داد که در آن  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودیها،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه خروجیها و  $c \equiv \{c_1, c_2, \dots, c_r\}$  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 [14]. اتوماتای یادگیر با ساختار متغیر<sup>۱۸</sup> توسط ۴ تایی  $\{\alpha, \beta, p, T\}$  نشان داده می‌شود که در آن  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عملهای اتوماتا،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه ورودیهای اتوماتا و  $p = \{p_1, p_2, \dots, p_r\}$  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

<sup>15</sup> Cellular Automata

<sup>16</sup> Learning Automata

<sup>17</sup> Cellular Learning Automata (CLA)

<sup>18</sup> Variable Structure Learning Automata

احتمالات برابر با یک باقی بماند. فرمولهای (۴) و (۵) یکی از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر را نشان می دهد.

الف- پاسخ مطلوب برای عمل  $i$ :

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

ب- پاسخ نامطلوب برای عمل  $i$ :

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

$a$  و  $b$  به ترتیب پارامتر پاداش و پارامتر جریمه می باشد. زمانی که  $a$  و  $b$  با هم برابر باشند، الگوریتم  $L_{R-P}$ <sup>۱۹</sup>، زمانی که  $a$  از  $b$  خیلی کوچکتر باشد، الگوریتم  $L_{R-EP}$ <sup>۲۰</sup> و زمانی که  $b$  مساوی صفر باشد، الگوریتم  $L_{R-I}$ <sup>۲۱</sup> نامیده می شود.

**اتوماتای سلولی** [15] از یک آرایه از سلولها با تعداد ابعاد محدود تشکیل شده است. هر سلول مجموعه ای محدود از حالات را می تواند اختیار کند. حالت بعدی هر سلول تابعی از حالت خود و همسایه هایش در حال حاضر می باشد. ساده ترین نوع اتوماتای سلولی یک بعدی می باشد که هر سلول آن حالت صفر و یا یک را می تواند انتخاب کند.

**اتوماتای یادگیر سلولی** ترکیبی از اتوماتای سلولی و اتوماتای یادگیر می باشد. در اتوماتای یادگیر سلولی [16] [17] پاداش و جریمه هر اتوماتا بستگی به حالت خود و همسایه هایش دارد. اتوماتای سلولی در [17] 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 می باشد. این شبکه می تواند یک شبکه متناهی، نیمه متناهی یا متناهی باشد.
- $\phi$  یک مجموعه متناهی از حالتها می باشد که هر سلول می تواند 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 می شود.
- $N = \{\bar{x}_1, \dots, \bar{x}_m\}$  یک زیر مجموعه متناهی از  $Z^d$  می باشد که بردار همسایگی نامیده می شود.
- $F: \phi^m \rightarrow \beta$  قانون محلی  $CLA$  می باشد به طوریکه  $\beta$  مجموعه مقادیری است که می تواند به عنوان سیگنال تقویتی پذیرفته شود.

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

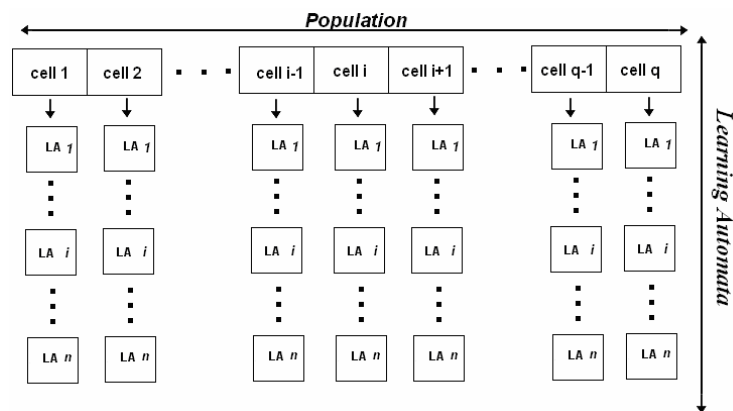
PSO از مجموعه ای از ذرها تشکیل شده که در فضای مسئله در حال پرواز هستند هر فرد در جمعیت دارای حافظه ای می باشد که بهترین مکانی را که آن فرد تا کنون در آنجا بوده ذخیره می کند و  $PB(Personal Best)$  نام دارد. افراد جمعیت بعد از هر جابجایی  $PB$  خود را در صورت نیاز به روز می کنند و برای حرکت بعدی به سمت  $PB$  خود و بهترین  $PB$  جمعیت  $GB(Global Best)$  سرعت می گیرند. در مدل  $CLA-PSO$  هر فرد یک سلول می باشد که به تعداد ابعاد دودویی مسئله دارای اتوماتای یادگیر می باشد. هر اتوماتای یادگیر در هر سلول متناظر با یکی از ابعاد دودویی مسئله می باشد 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 می شود. با توجه به اینکه در این مدل اتوماتاها دارای دو عمل می باشند، هر سلول با توجه به حالتش در یک رأس یک ابر

<sup>۱۹</sup>Linear Reward Penalty

<sup>۲۰</sup>Linear Reward Epsilon Penalty

<sup>۲۱</sup>Linear Reward Inaction

مکعب قرار می گیرد. شکل 2 چینش اتوماتای یادگیر را در یک اتوماتای سلولی یک بعدی که در CLA-PSO به کار رفته نشان می دهد.



شکل 2 چینش اتوماتاهای یادگیر در سلولها در توپولوژی یک بعدی

هر سلول دارای تعدادی همسایه می باشد و این همسایگی متقارن می باشد. همسایه های هر سلول در CLA-PSO برابر با سلول سمت چپ و راست آن می باشد. در این مدل، فرد  $i$  جمعیت بهترین جوابی را که تا کنون یافته در متغیری به نام  $PB_i$  ذخیره می کند. در طول اجرا، هر اتوماتای یادگیر در هر سلول با توجه به بردار احتمالات خود عملی (۰ یا ۱) را انتخاب می کند. از کنار گذاشتن این انتخابها، حالت هر سلول (ذره) جمعیت تعیین می شود. چنانچه این نقطه ی جدید در فضای مسئله دارای شایستگی<sup>۲۲</sup> بیشتری باشد،  $PB$  به روز می شود. 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 می باشد) برابر باشد، اتوماتای یادگیر پاداش می بیند و در غیر این صورت جریمه می شود. بعد از هر یادگیری احتمال یک بودن و صفر بودن هر اتوماتای یادگیر با متغیری به نام  $P_{max}$  مقایسه می شود. چنانچه یکی از این احتمالات از این پارامتر بزرگتر بود برابر  $P_{max}$  قرار داده می شود و دیگری طوری تغییر می کند که جمع احتمالات برابر با یک باقی بماند. جدول 2 شبه کد این الگوریتم را نشان می دهد.

```

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

```

جدول 2 شبه کد الگوریتم CLA-PSO

## ۵- تنظیم پارامترهای خوشه بندی مورچه ای به کمک CLA-PSO

همانطور که دیدیم، خوشه بندی مورچه ای دارای پارامترهای زیادی می باشد که تنظیم آنها برای کارایی بالا بسیار دشوار می باشد. بعضی از این پارامترها عبارتند از  $K_{pick}$ ،  $K_{drop}$ ،  $step\_size$  و  $\alpha$  و  $\sigma$ . در این مقاله از CLA-PSO برای تطبیق این پارامترها در خوشه بندی مورچه ای استفاده کرده ایم. ابعاد صفحه ای که خوشه بندی بر روی آن انجام می شود، طبق توصیه [3] برابر  $\sqrt{10 N_{data}} \times \sqrt{10 N_{data}}$  قرار داده شده است.

<sup>22</sup> Fitness

در این مقاله از دو نوع مورچه برای خوشه بندی استفاده کرده ایم که هرکدام نیمی از زمان اجرای خوشه بندی را در اختیار دارند و از CLA-PSO برای تطبیق پارامترهای این دو نوع مورچه استفاده کرده ایم. برای هر نوع از این مورچه ها ۵ پارامتر  $K_{pick}$ ,  $K_{drop}$ ,  $\alpha$  و  $step\_size$  را تنظیم کرده ایم، بنابراین ۱۰ پارامتر برای این دو نوع مورچه باید تنظیم شوند. ذره های CLA-PSO رشته های دودویی هستند که نشان دهنده یک ترکیب از این ۱۰ پارامتر برای تنظیم می باشد. جدول 3 نشان دهنده ی تعداد بیت های استفاده شده برای کد کردن پارامتر به شکل دودویی و حدود آنهاست.

جدول 3 تعداد بیت های 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 بیت ها	4	3	4	4	4

برای ارزیابی شایستگی هر ذره، رشته باینری آن ذره به ۱۰ پارامتر دهنده ی تبدیل شده و از آن پارامترها برای عمل خوشه بندی استفاده می شود. شایستگی هر ذره برابر با کیفیت خوشه بندی در نظر گرفته می شود. بعد از ارزیابی هر ذره مراحل CLA-PSO اجرا شده و نسل به نسل ذره ها (پارامترهای خوشه بندی) بهبود داده می شوند. تعداد جمعیت را برابر با ۳۰ و تعداد نسل ها را برابر با ۱۰۰ قرار داده ایم. برای تنظیم توسط الگوریتم ژنتیک، به شکل مشابه عمل شده با این تفاوت که از یک نوع مورچه برای خوشه بندی استفاده شده است [13]. بنابر این افراد جمعیت الگوریتم ژنتیک نشان دهنده ی ۵ پارامتر می باشند. برای ارزیابی خوشه بندی ذره ها در هنگام اجرای CLA-PSO و افراد در الگوریتم ژنتیک از مجموعه داده ای ۴ کلاسه با توزیع نرمال برای 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 می باشد. کیفیت خوشه بندی با استفاده از فرمول زیر تعیین می شود.

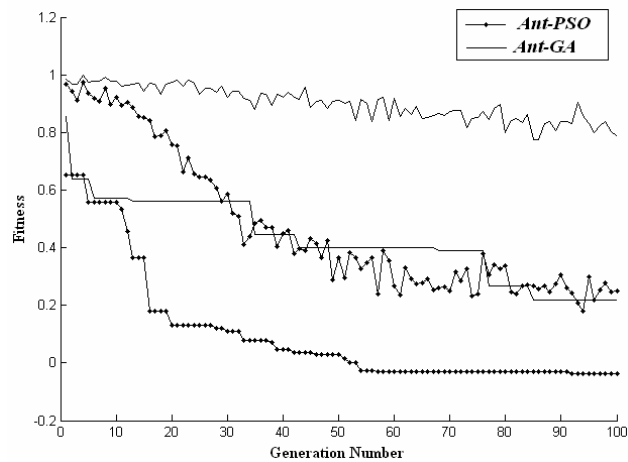
$$f_i = \sum_{\sigma \in \Sigma} [ \sum_{j \in Neigh(i, \sigma)} [-c(i, j) + 2(1 - c(i, j))] ] + (\sigma^2 - |Neigh(i, \sigma)|) \quad (6)$$

$$f = \sum_{i=1}^{N_{data}} f_i / N_{data}$$

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 می باشد.  $Neigh(i, \sigma)$  نشان دهنده ی مجموعه داده های همسایه های داده  $i$  در همسایگی  $\sigma$  بر روی صفحه خوشه بندی می باشد و  $|\Sigma| = \{3, 5, 7, 9, 11\}$  برای محاسبه ی  $f_i$ ، کلاس داده ی  $i$  با کلاسهای داده های همسایه مقایسه می شود. چنانچه کلاسها برابر نباشند عدد  $2 + (2(1 - c(i, j)))$  و چنانچه برابر باشد عدد  $1 - c(i, j)$  و فضای خالی عدد  $1 + (\sigma^2 - |Neigh(i, \sigma)|)$  در نظر گرفته می شود و با  $f_i$  جمع می شود. 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 می شود. 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 می باشد. این کاهش برابر با افزایش فضای خالی و نصف افزایش داده با کلاس متفاوت می باشد.  $f_i$  برای مقادیر مختلف  $\sigma$  محاسبه می شود. دلیل اینست که  $\sigma$  های کوچک، چگالی محلی را ارزیابی می کنند و  $\sigma$  های بزرگ جدایی خوشه ها را بر روی صفحه ارزیابی کند و ترکیب این دو می تواند به خوبی کیفیت خوشه بندی را ارزیابی کند.

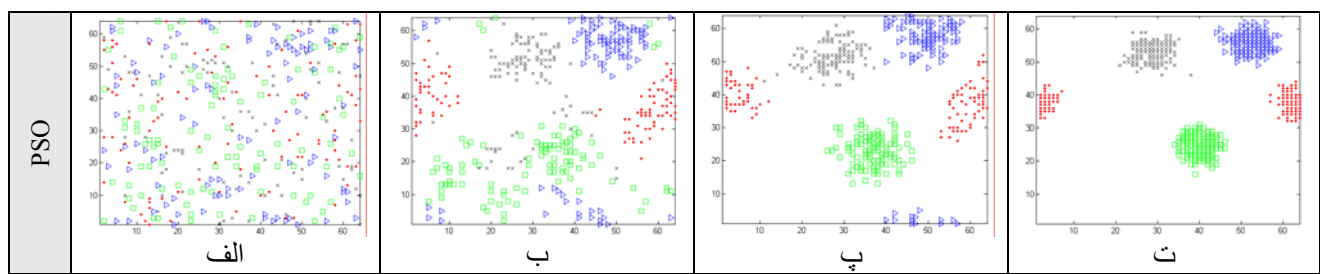
همانطور که از جدول 3 مشخص است هر فرد در الگوریتم ژنتیک دارای ۱۹ بیت و در CLA-PSO دارای  $(2 \times 19) = 38$  بیت می باشد. CLA مورد استفاده در CLA-PSO یک بعدی می باشد و اتوماتای یادگیر از قانون یادگیری  $L_{RP} (a = b = 0.1)$  استفاده می کند و  $P_{max}$  برابر با ۰/۹۵ قرار داده شده است. الگوریتم ژنتیک استفاده شده در [13] به این شکل می باشد: ۱۰٪ از بهترین افراد در هر نسل بدون تغییر به نسل بعد منتقل می شوند. برای بازترکیبی ۲ نفر از ۱۰٪ برتر به شکل اتفاقی انتخاب شده و یک فرزند تولید می شود. هر کدام از بیت های فرزند به صورت اتفاقی از یکی از والدین انتخاب می شود. برای جهش نیز هر بیت با

احتمال ۰/۰۱ تغییر می کند. شکل ۳ میانگین و بهترین شایستگی را در بین افراد جمعیت CLA-PSO و الگوریتم ژنتیک را نشان می دهد. در هر دو روش تعداد مورچه برابر با ۱۰ قرار داده شده اند.

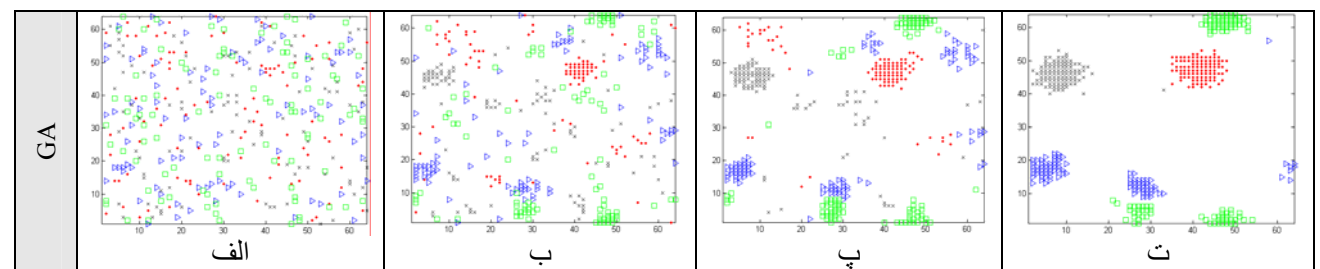


شکل ۳ میانگین و بهترین شایستگی در بین افراد جمعیت CLA-PSO و الگوریتم ژنتیک

همانطور که می بینید، میانگین شایستگی و بهترین شایستگی، با استفاده از CLA-PSO بهتر از الگوریتم ژنتیک می باشد. شکل ۴ و شکل ۵ مراحل خوشه بندی توسط پارامترهای بدست آمده توسط CLA-PSO (Ant-PSO) و الگوریتم ژنتیک (Ant-GA) را نشان می دهد.



شکل ۴ مراحل خوشه بندی Ant-PSO



شکل ۵ مراحل خوشه بندی Ant-GA

همانطور که شکل های ۴ و ۵ نشان می دهند، شیوه ی خوشه بندی پارامترهای بدست آمده توسط CLA-PSO با پارامترهای الگوریتم ژنتیک متفاوت می باشد. در Ant-PSO مورچه های نوع اول خوشه بندی خام تولید می کنند (شکل ۴ الف، ب) و پس از آن مورچه های نوع دوم خوشه های خام تشکیل شده توسط مورچه های نوع اول را پالایش می کنند (شکل ۴ پ، ت). مقایسه ی شکل ۴ و شکل ۵ نشان می دهد که پارامترهای بدست آمده توسط CLA-PSO بهتر از الگوریتم ژنتیک عمل می کنند. پارامترهای بدست آمده توسط CLA-PSO و الگوریتم ژنتیک را می توانید در جدول ۴ ببینید.



جدول 4 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 خوشه بندی 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 می کنیم و آن را به صورت آزمایش و خطا بدست می آوریم. برای تعیین  $\alpha$  مورچه ی دوم در Ant-PSO آن را  $0.8151$  ( $\frac{\alpha_{Ant2}}{\alpha_{Ant1}} = \frac{0.375}{0.46} = 0.8151$ ) برابر  $\alpha$  مورچه ی اول قرار می دهیم.  $step\_size$  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 آزمایش ها برابر با  $2\sqrt{10N_{data}}$  قرار داده ایم.

برای ارزیابی خوشه بندی انجام شده توسط روش ارائه شده (Ant-PSO) و روش موجود (Ant-GA) و K-means از ۴ معیار ارزیابی برای خوشه بندی ۵ داده استفاده کرده ایم. ۴ معیار مورد استفاده عبارتند از :  
**۱. F-Measure**: این معیار دو مفهوم دقت<sup>۲۳</sup> و به یاد آوری<sup>۲۴</sup> که از مفاهیم بازیابی اطلاعات هستند، استفاده می کند. این معیار و مقداری بین ۰ و ۱ دارد که مقادیر نزدیک به ۱ این معیار مطلوب هستند.

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

$$F(i, j) = \frac{(b^2 + 1). p(i, j). r(i, j)}{b^2 . p(i, j) + r(i, j)} \quad (7)$$

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

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  $j$  (بدست آمده توسط الگوریتم خوشه بندی) و تعداد داده های کلاس  $i$  در خوشه  $j$   $p$  و  $r$  نیز به ترتیب نشان دهنده ی مفاهیم دقت و یادآوری می باشند. 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 خوشه بندی) و  $U$  آرایه باشد که کلاس داده ها را نشان می دهد. این معیار به این شکل تعریف می شود و مقداری بین ۰ و ۱ دارد. مقادیر نزدیک به ۱ این معیار مطلوب هستند.

$$a = \{i, j | U(i) = U(j) \wedge V(i) = V(j)\}$$

$$b = \{i, j | U(i) = U(j) \wedge V(i) \neq V(j)\}$$

$$c = \{i, j | U(i) \neq U(j) \wedge V(i) = V(j)\}$$

$$d = \{i, j | U(i) \neq U(j) \wedge V(i) \neq V(j)\} \quad (8)$$

$$R = \frac{a + d}{a + b + c + d}$$

<sup>23</sup> Precision

<sup>24</sup> Recall

۳. **Entropy**: این پارامتر اعدادی بین ۰ و ۱ تولید می کند و اعداد نزدیک به صفر مطلوب می باشند.

$$p_{ij} = I(i, j) / n_i$$

$$e_i = - \sum_j p_{ij} \log p_{ij} \quad (9)$$

$$E = \sum_i e_i n_i / n$$

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

۴. **واریانس درون کلاسی (Variance)**:

$$V = \sum_{c \in C} \sum_{i \in c} d(\mu_c, i)^2 \quad (10)$$

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

۵ داده ی مورد استفاده عبارتند از: **Iris.۱**: دارای ۱۵۰ داده ی ۴ بعدی می باشد که در ۳ کلاس به شکل مساوی تقسیم شده اند. **Wisconsin.۲**: از ۶۸۳ داده ی ۱۰ بعدی تشکیل شده و دارای ۲ کلاس با تعداد داده های ۲۳۹ و ۴۴۴ می باشد. **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 می باشد. **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 معیار ۲ تقسیم شده اند. **Glass.۵** که شامل ۲۱۴ داده ی ۹ بعدی است و به ۶ کلاس که دارای ۷۶، ۷۰، ۲۹، ۱۷، ۱۳ و ۹ داده می باشند 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 ۱ داشته 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 فاصله 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 ۵ ذکر شده اند

جدول ۵ پارامترهای مورد استفاده در خوشه بندی 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 ۷ میانگین ۴ معیار را برای ۵ داده در ۲۰ تکرار گزارش می کند.

جدول ۶ مقایسه ی خوشه بندی 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
<b>Ecoli</b>	<b>k-means</b>	<b>Ant-GA</b>	<b>Ant-PSO</b>
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
<b>Glass</b>	<b>k-means</b>	<b>Ant-GA</b>	<b>Ant-PSO</b>
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

جدول 7 مقایسه ی خوشه بندی روشهای Ant-PSO و Ant-GA و Kmeans برای ۵ داده و ۴ معیار

نتایج جدول 7 نشان می دهد 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 می کند. جدول بالا همچنین کارایی بالای روش ارائه شده را در مقایسه با K-Means نشان می دهد.

## ۷- نتیجه گیری

در این مقاله از CLA-PSO برای تطبیق پارامترهای خوشه بندی مورچه ای استفاده کردیم. برای خوشه بندی 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 می کند مقایسه کردیم. مقایسه ها نشان می دهد که روش ارائه شده پارامترهایی تولید می کند که کارایی بالایی در مقایسه با پارامترهای بدست آمده توسط الگوریتم ژنتیک و روش خوشه بندی K-means دارند.

## مراجع

- [1] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "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.
- [2] E. D. Lumer and B. Faieta. "Diversity and adaptation in populations of clustering ants", Proc. of the Third International Conference on The Simulation of Adaptive Behavior: From Animals to Animats 3, MIT Press, pp. 449–508, 1994.
- [3] J. Handl, J. Knowles, and M. Dorigo. "Strategies for the increased robustness of ant-based clustering", Lecture Notes in Computer Science, Vol. 2977, pp. 90–104, 2004.
- [4] J. Handl and B. Meyer. "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.
- [5] K. Hoe, W. Lai, and T. Tai. "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.
- [6] P. Kuntz and D. Snyers. "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.
- [7] P. Kuntz, D. Snyers, and P. Layzell. A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning. Journal of Heuristics, 5(3):327–351, 1998.
- [8] P. Kuntz, P. Layzell, and D. Snyers, "A colony of ant-like agents for partitioning in VLSI technology", Proceedings of 4th European Conference on Artificial Life, MIT Press, July 1997.
- [9] Kennedy, J., and Eberhart, R. C., "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, , pp. 1942–1948, 1995.
- [10] 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.
- [11] 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.

- [12] B. Jafarpour, M. R. Meybodi, "*A Hybrid Method for Optimization (Discrete PSO + CLA)*", Accepted in IEEE International Conference on Intelligent and Advanced Systems, Malaysia, 2007.
- [13] C. Aranha, H. Iba, "*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.
- [14] M. A. L. Thathachar, P. S. Sastry, "*Varieties of Learning Automata: An Overview*", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.
- [15] Wolfram, S., *Cellular Automata and Complexity*, Perseus Books Group, 1994.
- [16] 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.
- [17] 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.
- [18] Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989