

SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



عضویت در خبرنامه



فیلم های آموزشی

کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی



PROPOSAL

پروپوزال

مركز آموزش
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین
پروپوزال نویسی و پایان نامه نویسی



مركز آموزش
روش تحقیق و مقاله نویسی علوم انسانی

کارگاه آنلاین
روش تحقیق و مقاله نویسی علوم انسانی



ISI
Scopus

مركز آموزش
آشنایی با پایگاه های اطلاعات علمی بین المللی و ترکیه های جستجو

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترکیه های جستجو

دو الگوریتم تقریبی جدید برای پیدا کردن گراف میانه عمومی

کریم فائز رضا رستگار

آزمایشگاه پردازش تصویر و شناسایی الگو - دانشکده برق - دانشگاه صنعتی امیرکبیر - تهران - ایران

kfaez@aut.ac.ir

rrastegar@ce.aut.ac.ir

چکیده: گراف میانه عمومی¹، یک مفهوم مناسب برای استخراج ویژگیهای مفیدی از مجموعه ای از گراف ها می باشد که علاوه بر استفاده از آن بعنوان نمونه برای مجموعه، می تواند در راستای گسترش مفاهیم و الگوریتمهای استفاده شده در شناسایی آماری الگو به شناسایی ساختاری الگو، مورد استفاده قرار گیرد. از طرفی محاسبه این گراف از نظر زمانی و محاسباتی بسیار پیچیده است و همین امر، لزوم ایجاد الگوریتمهای تقریبی که در زمان معقول ما را به نتیجه برسانند، آشکار می سازد. ما در این مقاله در ابتدا مفهوم میانگین دو گراف صفت دار را که اخیراً معرفی شده است، به مجموعه ای با تعداد بیش از دو گراف صفت دار^{II} تعمیم داده و الگوریتمی برای محاسبه آن ارائه می دهیم و با استفاده از آن گراف میانه عمومی را تقریب می زنیم. سپس در ادامه مفهوم جدید و کلی تر دیگری با نام میانگین وزن دار را معرفی و روشی برای محاسبه آن پیشنهاد می دهیم. در قسمت دیگر مقاله الگوریتم جدید دیگری با استفاده از الگوریتم تخمین توزیع^{III} UMDA^{IV} برای محاسبه گراف میانه عمومی ارائه خواهیم کرد که در عین کارایی، از سادگی قابل توجهی برخوردار بوده و بار محاسباتی کمی نیز دارا می باشد.

کلمات کلیدی: گراف میانه عمومی، گراف صفت دار، الگوریتم UMDA.

۱- مقدمه

خوشه بندی یک ابزار کلیدی در شناسایی الگو می باشد و در حال حاضر تعداد بسیاری از الگوریتمهایی که بر اساس ویژگی های برداری به خوشه بندی فضای الگوها می پردازند، در شناسایی آماری الگو بوجود آمده اند. اما با توجه به این که شناسایی ساختاری الگو نیز تواناییهای بسیاری برای شناسایی الگوهای با ساختار پیچیده دارد، توجه اندکی به ایجاد الگوریتمهای مشابه در حوضه ساختارهای سمبولیک شده است. یکی از این رهیافتهایی که در حال حاضر برای افزایش تواناییهای ساختارهای سمبولیک در بازنمایی الگوها و جستجو در میان آنها، مورد توجه قرار گرفته است، محاسبات میانی می باشد. محاسبات میانی در فضای رشته ای چندی است که مورد مطالعه قرار گرفته و الگوریتمهای متنوعی برای آن ارائه شده است. اما مفهوم میانه در فضای گرافی اخیراً در [2,1] پیشنهاد شده است. گراف میانه با تمام ویژگیهای سودمند، به راحتی قابل محاسبه نیست و با افزایش تعداد گرافها در مجموعه و بزرگ شدن آنها از لحاظ تعداد نود و یال، دسترسی به گراف میانه بسیار سخت و زمانگیر می شود. همین امر لزوم ایجاد الگوریتمهای سریعتری را برای محاسبه گراف میانه، آشکار می سازد. تا کنون تعداد اندکی الگوریتم برای محاسبه گراف میانه با استفاده از الگوریتم ژنتیکی [2] و الگوریتم های تقریبی^V [3] ارائه شده است که بعضاً برای گرافهای بسیار خاص می باشند. ما در این مقاله، دو الگوریتم تقریبی جدید را برای محاسبه گراف میانه ارائه می دهیم. در الگوریتم اول، ابتدا مفهوم میانگین وزن دار دو گراف صفت دار را که در [4] معرفی شده است، به مجموعه ای با بیش از دو گراف صفت دار تعمیم می دهیم سپس گراف میانه عمومی را با محاسبه گراف میانگین برای مجموعه گراف، تخمین می زنیم. البته تعمیم میانگین وزن دار برای فضای گرافی می تواند امکان مطالعه آماری بروی فضای گرافی را فراهم سازد که ما در این مقاله از این ویژگی آن چشم پوشی کرده ایم. در ادامه مقاله الگوریتم دیگری برای محاسبه گراف میانی پیشنهاد داده ایم. در این الگوریتم مبنای کار بر محاسبات تکاملی می باشد. نوع الگوریتم تکاملی که از آن استفاده کرده ایم، الگوریتم تخمین توزیع می باشد که اخیراً به منظور رفع مشکلات الگوریتمهای ژنتیکی مطرح شده است. انتخاب الگوریتم تخمین توزیع برای محاسبه گراف میانه عمومی برای محک زدن این رهیافت در حل مسائل با پیچیدگی بسیار بالا و ارائه یک الگوریتم ساده و سریع برای محاسبه گراف میانه عمومی بوده است.

¹ Generalized Median Graph

^{II} Attributed Graph

^{III} Estimation of Distribution Algorithm

^{IV} Univariate Marginal Distribution Algorithm

^V Approximation Algorithms

آنچه در ادامه خواهد آمد به این شرح است. در بخش ۲ به ارائه تعریف رسمی گراف صفت دار، فاصله بین دو گراف صفت دار، گراف میانه مجموعه^۱ و گراف میانه عمومی می پردازیم. سپس در قسمت ۳ گراف میانگین را برای یک مجموعه گراف بدست می آوریم و با استفاده از آن تخمینی برای گراف میانه عمومی می یابیم و در پایان بخش ۳ گراف میانگین را به گراف میانگین وزن دار توسعه داده و برای محاسبه آن الگوریتمی ارائه می دهیم. در ادامه مقاله در بخش ۴ به تشریح الگوریتم تکاملی UMDA پرداخته و در بخش ۵ گراف میانه عمومی را با استفاده از این الگوریتم محاسبه می کنیم. و در نهایت با ارائه نتایج شبیه سازی، مقایسه و نتیجه گیری، مقاله را به پایان می بریم.

۲- تعاریف رسمی

تعریف ۱- فرض کنید یک گراف صفت دار $G = \langle V, E, \mu, \xi \rangle$ با یک چهارتایی $G = \langle V, E, \mu, \xi \rangle$ تعریف شود به نحوی که [6, 5, 4]:

- V مجموعه نودها است.

- $E \subseteq V \times V$ مجموعه یالها می باشد.

- $\mu: V \rightarrow L_V$ تابعی است که برچسبها را به نودها منتسب می کند.

- $\xi: E \rightarrow L_E$ تابعی است که برچسبها را به یالها منتسب می کند.

باید توجه داشت که در تعریف فوق گراف، یالها جهت دار و نودها و یالها دارای برچسب می باشند و برای مثال اگر داشته باشیم

$(a, b) \in E, a, b \in V$ ، انگاه در حالت عمومی رابطه $(b, a) \notin E$ بر قرار است.

در ادامه کار به تعریف ایزومورفیزم و شباهت بین دو گراف می پردازیم.

تعریف ۲- یک تابع معکوس پذیر $f: V \rightarrow V'$ ، یک ایزومورفیزم از گراف $G = \langle V, E, \mu, \xi \rangle$ به

گراف $G' = \langle V', E', \mu', \xi' \rangle$ می باشد اگر و فقط اگر:

- برای تمام $v \in V$ داشته باشیم $\mu(v) = \mu'(f(v))$.

- برای تمام یالهای $e = (v_1, v_2) \in E$ ، یک یال $e' = (f(v_1), f(v_2)) \in E'$ وجود داشته باشد که $\xi'(e') = \xi(e)$ و برای

هر $e' = (v'_1, v'_2) \in E'$ یک یال $e = (f^{-1}(v'_1), f^{-1}(v'_2)) \in E$ موجود باشد بطوریکه $\xi'(e') = \xi(e)$.

ایزومورفیزم یک مفهوم سودمند برای تعیین شباهت بین دو گراف است. از آنجایی که معمولا محیطهای واقعی با نویز همراه هستند، دو گراف با وجود شباهتهای بسیار زیاد، کاملا با هم مساوی نیستند. به همین دلیل ایجاد مفهوم فاصله بین دو گراف لازم می باشد. معیارهای متفاوتی برای تعیین فاصله بین دو گراف مطرح شده اند، که ما در اینجا به یکی از معروفترین آنها که بر پایه عملگرهای ویرایشی تعریف می شود، استناد می کنیم.

تعریف ۳- یک گراف $G = \langle V, E, \mu, \xi \rangle$ داده شده است. یک عملگر ویرایشی گراف، δ ، بصورت زیر تعریف می شود:

- اگر $l \in L_V, v \in V$ ، انگاه $l \rightarrow v$ یک جایگزینی برچسب $\mu(v)$ با برچسب l می باشد.

- اگر $l' \in L_E, e \in E$ ، انگاه $l' \rightarrow e$ یک جایگزینی برچسب $\xi(e)$ با برچسب l' می باشد.

- اگر $v \in V$ ، انگاه $v \rightarrow \lambda$ پاک کردن نود v از گراف G است. باید توجه داشت که v یک نود ایزوله-نودی که با هیچ نودی یال مشترک ندارد- است.

- اگر $v \notin V$ ، انگاه $v \rightarrow \lambda$ اضافه کردن نود v با برچسب $l \in L_V$ به گراف G است.

- اگر $e \in E$ ، انگاه $e \rightarrow \lambda$ ، پاک کردن یال e از گراف G می باشد.

- اگر $e \notin E$ ، انگاه $e \rightarrow \lambda$ ، اضافه کردن یال $e = (v_1, v_2)$ با برچسب $l \in L_E$ به گراف G است. باید توجه داشت که عملگر $\lambda \rightarrow e$ در صورت $v_1, v_2 \in V$ مجاز می باشند.

این شش عملگر تعریف شده برای تبدیل هر گراف G به G' مناسب و کافی است و با استفاده از همین عملگرها فاصله بین دو گراف اندازه گیری می شود. به این منظور در ابتدا بر اساس کاربرد به هر یک از عملگرهای δ هزینه $C(\delta)$ را انتساب می دهیم. معمولا $C(\delta) \in [0, \infty)$ می باشد. البته در بعضی کاربردها (پردازش بر روی رشته های ژنی) این شرط نقض می شود.

تعریف ۴- یک گراف $G = \langle V, E, \mu, \xi \rangle$ و عملگر ویرایشی δ در نظر بگیرید. $\delta(G)$ یک

گراف $\delta(G) = \langle V_\delta, E_\delta, \mu_\delta, \xi_\delta \rangle$ است بطوریکه:

¹ Set Median Graph

$$V_\delta = \begin{cases} V \cup \{v\} & \text{iff } \delta = (\lambda \rightarrow v) \\ V - \{v\} & \text{iff } \delta = (v \rightarrow \lambda) \\ V & \text{otherwise} \end{cases} \quad (1)$$

$$E_\delta = \begin{cases} E \cup \{e\} & \text{iff } \delta = (\lambda \rightarrow e) \\ E - \{e\} & \text{iff } \delta = (e \rightarrow \lambda) \\ E \cap (V_\delta \times V_\delta) & \text{otherwise} \end{cases} \quad (2)$$

$$\mu_\delta(v) = \begin{cases} l & \text{iff } \delta = (\mu(v) \rightarrow l) \text{ or } \delta = (\lambda \rightarrow v) \\ \mu(v) & \text{otherwise} \end{cases} \quad (3)$$

$$\xi_\delta(e) = \begin{cases} l & \text{iff } \delta = (\xi(e) \rightarrow l) \text{ or } \delta = (\lambda \rightarrow e) \\ \xi(e) & \text{otherwise} \end{cases} \quad (4)$$

تعریف ۵- یک گراف صفت دار $G = \langle V, E, \mu, \xi \rangle$ و یک توالی از عملگرهای $\Delta = (\delta_1, \delta_2, K, \delta_k)$ داده شده اند. گراف ویرایش شده $\Delta(G)$ گرافی است که:

$$\Delta(G) = \delta_k(K(\delta_2(\delta_1(G))K)) \quad (5)$$

و هزینه تبدیل گراف G به $\Delta(G)$ برابر است با:

$$C(\Delta) = \sum_{i=1}^k C(\delta_i) \quad (6)$$

با توجه به محدودیتهایی که در اعمال عملگرها بروی گرافها وجود دارد، این امکان وجود دارد که $\Delta(G)$ برای یک Δ خاص وجود نداشته باشد. در ادامه کار هر جا از Δ استفاده کردیم، فرض می کنیم $\Delta(G)$ وجود داشته باشد.

تعریف ۶- دو گراف G, G' داده شده اند. یک ایزومورفیسم تصحیح-خطا گراف^۱ φ با نام اختصاری ecgi، از گراف G به G' ، یک دوتایی $\varphi = (\Delta, f_\Delta)$ می باشد که Δ یک توالی از عملگرهای ویرایشی و f_Δ یک ایزومورفیسم از $\Delta(G)$ به G' است. هزینه φ برابر با هزینه Δ تعریف می شود و داریم $C(\varphi) = C(\Delta)$. نکته بسیار مهم این است که معمولا بیشتر از یک Δ می توان یافت که به ازای آن یک ایزومورفیسم f_Δ از $\Delta(G)$ به G' وجود داشته باشد. بنابراین با توجه به یکتا نبودن φ بین دو گراف، تعریف زیر را برای فاصله دو گراف ارائه می دهیم.

تعریف ۷- فرض کنید G, G' دو گراف باشند. فاصله تصحیح-خطا بین G, G' که با $d(G, G')$ نمایش داده می شود، به صورت زیر تعریف می گردد:

$$d(G, G') = \arg \min_{\Delta} \{C(\Delta) | \varphi = (\Delta, f_\Delta) \text{ is an ecgi from } G \text{ to } G'\} \quad (7)$$

باید توجه داشته باشید که در حالت کلی $d(G', G) \neq d(G, G')$

حال با توجه به تعریف فاصله بین دو گراف به تعریف مفاهیم گراف میانی مجموعه و گراف میانی عمومی می پردازیم.

تعریف ۸- فرض کنید $S = \{G_i = \langle V_i, E_i, \mu_i, \xi_i \rangle | 1 \leq i \leq n\}$ یک مجموعه از n گراف G_i باشد. آنگاه گراف میانه مجموعه \hat{G} ، به صورت زیر تعریف می شود [4]:

$$\hat{G} = \arg \min_{G \in S} \frac{1}{n} \sum_{i=1}^n d(G, G_i) \quad (8)$$

و به همین صورت گراف میانه عمومی \bar{G} ، تعریف می شود:

$$\bar{G} = \arg \min_{G \in U} \frac{1}{n} \sum_{i=1}^n d(G, G_i) \quad (9)$$

بطوریکه U مجموعه تمام گرافهایی که می توان با استفاده از L_V, L_E ساخت، می باشد.

گراف میانی عمومی از گراف میانی مجموعه، مفهومی سودمندتر و پرکاربردتر است و برای استفاده به عنوان نمونه در یک کلاس از گرافها مناسبتر از گراف میانی مجموعه می باشد. بدست آوردن هر دو گراف برای یک مجموعه یک مسئله ذاتا دشوار^{II} است. دلیل پیچیدگی آن در نگاه

^I Error Correction Graph Isomorphism

^{II} NP-Complete

اول بدلیل استفاده از مفهوم فاصله بین دو گراف، که خود از مسایل بفرنج و نمایی است، می باشد. گراف میانی عمومی، \bar{G} ، علاوه بر پیچیدگی فوق از پیچیدگی U ، که اندازه آن تابع نمایی از $|L_E|$ یا $|L_V|$ می باشد، نیز رنج می برد. به دلیل اهمیت بدست این گراف در پر کردن شکاف ایجاد شده بین شناسایی آماری الگو و شناسایی ساختاری الگو، بدست آوردن الگوریتمهای تقریبی با زمان پردازشی معقول و در عین حال خروجی تقریباً بهینه، کاری بسیار ارزشمند است.

۳- گراف میانگین و تخمین گراف میانه عمومی بوسیله آن

همانطور که قبلاً نیز اشاره کردیم، بسط الگوریتمهای استفاده شده در شناسایی آماری الگو به شناسایی ساختاری الگو می تواند سبب حل بسیاری از ناتوانیهای موجود در شناسایی الگو باشد. یکی از مفاهیمی که پایه و اساس محاسبات بروی ویژگیهای برداری الگوها، در شناسایی آماری الگو میباشد، مفهوم میانگین است. شاید دور از واقعیت نباشد که بگوییم، کلید ورودی به دنیای آمار و احتمال و استفاده از قدرت این علم در شناسایی الگو همین مفهوم میانگین است. محاسبه میانگین در فضای برداری کاری ساده می باشد. در [4,5] میانگین وزن دار برای دو گراف صفت دار معرفی شده است. هدف ما در این بخش بدست آوردن آن برای یک مجموعه از گرافها و تقریب گراف میانی عمومی بوسیله آن است.

فرض کنید $F = \{v_i^0 \mid 1 \leq i \leq n, v_i^0 \in \mathfrak{R}^m\}$ باشد. برای مجموعه F ، میانگین حسابی M_n^0 با استفاده از فرمول زیر محاسبه می شود:

$$M_n^0 = \frac{1}{n} \sum_{i=1}^n v_i^0 \Rightarrow M_n^0 = \frac{(n-1)}{n(n-1)} \sum_{i=1}^{n-1} v_i^0 + \frac{1}{n} v_n^0 \Rightarrow M_n^0 = (1 - \frac{1}{n}) M_{n-1}^0 + \frac{1}{n} v_n^0 \quad (10)$$

$$\Rightarrow (M_n^0 - M_{n-1}^0) = \frac{1}{n} (v_n^0 - M_{n-1}^0)$$

همانطور که دیده می شود، با کمی تغییرات بروی فرمول میانگین در فضای برداری آن را به صورت بازگشتی درآوردیم. حال اگر به جای $(a^0 - b^0)$ در فرمول فوق از $d(a^0, b^0) = (a^0 - b^0)$ استفاده کنیم، عبارت به صورت زیر درمی آید:

$$d(M_n^0, M_{n-1}^0) = \frac{1}{n} d(v_n^0, M_{n-1}^0) \quad (11)$$

در این مرحله توجه شما را به یک قضیه در مورد فاصله در فضای گرافی جلب می کنیم.

قضیه ۱- فرض کنید دو گراف، $G, G' = (\Delta, f_\Delta)$ یک ecgi بهینه از G به G' ، Δ' یک زیر دنباله از Δ و $\alpha = C(\Delta')$ باشد. آنگاه می توانیم یک گراف G'' را بگونه ای تولید کنیم که داشته باشیم [4]:

$$d(G, G'') = \alpha \quad (12)$$

$$d(G, G') = \alpha + d(G'', G') \quad (13)$$

بنابراین با توجه به قضیه (۱) می توانیم به ازای هر دو گراف داده شده G, G' ، گراف میانگین وزن دار آنها را با وزن $0 \leq \beta \leq 1$ بیابیم:

$$d(G, G') = \beta' d(G, G'') + (1 - \beta') d(G'', G') \quad (14)$$

$$\beta' = C(\arg \min_{\Delta'' \subseteq \Delta} |\beta - \frac{C(\Delta'')}{C(\Delta)}|) \quad (15)$$

همانطور که مشاهده می کنید به ازای مقادیر $\beta = \beta'$ ، گراف G'' دقیقاً میانگین وزن دار بین G, G' است. و در حالتیکه $\beta \neq \beta'$ می توان گراف وزن دار را به طور تقریبی بدست آورد. باید توجه داشته باشید که به ازای تمام $\Delta''(G)$ ، Δ'' ممکن است موجود نباشد. حال به محاسبه میانگین باز می گردیم و بجای $d(a^0, b^0)$ در فرمول (۱۱)، $d(G, G')$ را قرار میدهم. با فرض اینکه \tilde{G}_i گراف میانگین مجموعه $S_i = \{G_j \mid G_j = \langle V_j, E_j, \mu_j, \xi_j \rangle, 1 \leq j \leq i\}$ باشد، الگوریتم زیر را اجرا می کنیم:

الگوریتم ۱- یافتن گراف میانگین

ورودی: مجموعه گراف S_n .

خروجی: گراف میانگین \tilde{G}_n .

قدم ۱- قرار می دهیم: $G_1 = \tilde{G}_1$ و $i = 1$.

قدم ۲- ابتدا از گراف G_{i+1} به گراف \tilde{G}_i ، یک ایزومورفیسم تصحیح-خطای گراف بهینه $\varphi_i = (\Delta_i, f_{\Delta_i})$ پیدا می کنیم. سپس:

$$\tilde{\Delta}_i = \arg \min_{\Delta' \subseteq \Delta_i} |\frac{i C(\Delta_i)}{i+1} - C(\Delta')| \quad (16)$$

قدم ۳- قرار می دهیم: $\tilde{G}_{i+1} = \tilde{\Delta}_i(G_{i+1})$.

قدم ۴- $i = i + 1$

قدم ۵- اگر $i < n$ به گام ۲ می رویم والا خارج می شویم.

پیچیدگی الگوریتم فوق نسبت به عملگر φ ، $O(n)$ است. البته پیدا کردن $\varphi = (\Delta, f_\Delta)$ بهینه، خود یک مسئله ذاتا دشوار است که می تواند به روشهای متفاوت حل شود. همچنین یافتن $\tilde{\Delta}_i$ ، در مورد مسائل خاص ساده است، اما در حالت کلی یافتن آن دارای پیچیدگی $O(2^{|\Delta_i|})$ است. حال با پذیرفتن درصدی از خطا، گراف میانی عمومی را برابر \tilde{G}_n می گیریم. برای بهبود این فرض می توانیم الگوریتم (۱) را به ازای جایگشتهای مختلفی از S_n اجرا کنیم، سپس بهترین جواب را به عنوان \bar{G} بپذیریم. البته انتخاب جایگشتهای می تواند بر اساس ویژگیهای اعضای S_n و بصورت مکاشفه ای صورت پذیرد.

از مفهوم گراف میانگین می توانیم واریانس را برای مجموعه S_n حساب کنیم:

$$Var(S_n) = \frac{1}{n-1} \sum_{i=1}^n d^2(G_i, \tilde{G}_n) \quad (17)$$

و از واریانس محاسبه شده می توانیم برای اندازه گیری کیفیت الگوریتم خوشه بندی فضای گرافی، استفاده کنیم.

به طور مشابه می توان مفهوم میانگین وزن دار n بردار در فضای m بعدی را به فضای گرافی بسط داد. مجددا فرض کنید:

$$F = \{\tilde{v}_i^{\omega} | 1 \leq i \leq n, \tilde{v}_i \in \mathfrak{R}^m\} \quad (18)$$

برای مجموعه F ، میانگین وزن دار \tilde{W}_n با استفاده از فرمول زیر محاسبه می شود:

$$\begin{aligned} \tilde{W}_n &= \sum_{i=1}^n \alpha_i \tilde{v}_i / \sum_{i=1}^n \alpha_i \Rightarrow (\sum_{i=1}^n \alpha_i) \tilde{W}_n = \sum_{i=1}^{n-1} \alpha_i \tilde{v}_i + \alpha_n \tilde{v}_n \\ \Rightarrow (\sum_{i=1}^n \alpha_i) \tilde{W}_n &= (\sum_{i=1}^{n-1} \alpha_i) \tilde{W}_{n-1} + \alpha_n \tilde{v}_n \Rightarrow (\sum_{i=1}^{n-1} \alpha_i) (\tilde{W}_n - \tilde{W}_{n-1}) = \alpha_n (\tilde{v}_n - \tilde{W}_{n-1}) \\ \Rightarrow (\sum_{i=1}^{n-1} \alpha_i) d(\tilde{W}_n, \tilde{W}_{n-1}) &= \alpha_n d(\tilde{v}_n, \tilde{W}_{n-1}) \end{aligned} \quad (19)$$

و از طرفی:

$$d(\tilde{v}_n, \tilde{W}_n) + d(\tilde{W}_n, \tilde{W}_{n-1}) = d(\tilde{v}_n, \tilde{W}_{n-1}) \quad (20)$$

در نتیجه:

$$d(\tilde{v}_n, \tilde{W}_n) = (\sum_{i=1}^{n-1} \alpha_i / \sum_{i=1}^n \alpha_i) d(\tilde{v}_n, \tilde{W}_{n-1}) \quad (21)$$

که $d(\tilde{a}, \tilde{b})$ فاصله برداری دو بردار \tilde{a}, \tilde{b} است. برای گسترش مفهوم میانگین وزن دار به مجموعه گراف S_n ، بجای $d(\tilde{a}, \tilde{b})$ از $d(G, G')$ استفاده می کنیم. با توجه به قضیه (۱) و مراحل انجام شده در استخراج الگوریتم (۱) و با فرض اینکه \tilde{G}_i میانگین وزن دار مجموعه $S_i = \{G_j | 1 \leq j \leq i\}$ و α_i وزن مربوط به گراف G_i باشد، الگوریتم زیر را اجرا می کنیم.

الگوریتم ۲- یافتن گراف میانگین وزن دار

ورودی: مجموعه گراف S_n .

خروجی: گراف میانگین \tilde{G}_n .

قدم ۱- قرار می دهیم: $\tilde{G}_1 = G_1$ و $i = 1$.

قدم ۲- ابتدا از گراف G_{i+1} به گراف \tilde{G}_i ، یک ایزومورفیزم تصحیح-خطای گراف بهینه $\varphi_i = (\Delta_i, f_{\Delta_i})$ پیدا می کنیم و سپس:

$$\tilde{\Delta}_i = \arg \min_{\Delta' \subseteq \Delta_i} |(\sum_{j=1}^i \alpha_j / \sum_{j=1}^{i+1} \alpha_j) - (C(\Delta') / C(\Delta_i))| \quad (22)$$

قدم ۳- قرار می دهیم: $\tilde{G}_{i+1} = \tilde{\Delta}_i(G_{i+1})$.

قدم ۴- $i = i + 1$

قدم ۵- اگر $i < n$ به گام ۲ می رویم والا خارج می شویم.

کاملا واضح است که الگوریتم (۲) با فرض $\alpha_1 = K = \alpha_n = 1/n$ به الگوریتم (۱) تبدیل می شود.

۴- الگوریتم های تخمین توزیع

دسته بزرگی از الگوریتمهای بهینه سازی، الگوریتمهای تکاملی مانند الگوریتمهای ژنتیک می باشند. الگوریتمهای ژنتیکی به دلیل سادگی و عدم نیاز به معادلات دیفرانسیل پیچیده در حل مسائل بفرنج با فضای جستجوی غیر هموار، مانند تعیین گراف میانه عمومی و تطابق گرافها [6,7,8]، مورد استفاده قرار می گیرند. اما همانطور که می دانید رفتار الگوریتم ژنتیکی وابستگی شدیدی به پارامترهایی از جمله نحوه تعریف عملگرهای جهش و تولید نسل، احتمال جهش و تولید نسل، اندازه جمعیتها و تعداد نسلهای تولید شده، دارد. بر اساس کاربرد و حتی گاه برای ورودیهای متفاوت این پارامترها نیاز به تغییر دارند و جالب آنکه در بسیاری از موارد یافتن همین پارامترها به صورت یک مسئله بهینه سازی در می آید. با توجه به مطالب گفته شده، دسته ای از الگوریتمهای مکاشفه ای غیر قطعی مبتنی بر جمعیت با نام الگوریتمهای تخمین توزیع ارائه شده اند که همانند الگوریتمهای ژنتیکی نیازمند به فضای جستجوی هموار و معادلات دیفرانسیل پیچیده نمی باشند و علاوه بر این بسیاری از مشکلات الگوریتمهای ژنتیکی را در تنظیم پارامترها، حل نموده اند. در الگوریتمهای تخمین توزیع با ساخت یک مدل احتمالاتی از اجزای سازنده ژنوم، سرعت پیشروی به سوی جواب بهینه مسئله افزایش می یابد. همچنین جمعیتهای جدید با استفاده از عملگرهای جهش و تولید نسل ساخته نمی شوند، بلکه ژنومهای جدید، بر اساس توزیع احتمالاتی تخمین زده شده بر اساس ژنومهای انتخاب شده از نسلهای قبل، نمونه برداری و ساخته می شوند. در الگوریتمهای تخمین توزیع رابطه بین جوابها و اجزای آنها برخلاف سایر الگوریتمهای تکاملی به صورت صریح تخمین زده می شود و در واقع تخمین همین رابطه مشکلترین بخش الگوریتمهای تخمین توزیع است. غالب کلی الگوریتمهای تخمین توزیع به صورت زیر است [9,10]:

الگوریتم ۳- الگوریتم تخمین توزیع

قدم ۱- در ابتدا نخستین جمعیت، D_0 ، با تعداد N ژنوم تشکیل می شود. معمولاً تولید N ژنوم مذکور، با فرض توزیع یکنواخت بروی هر یک از متغیرها صورت می پذیرد. سپس هر یک از ژنومها ارزیابی می شوند.

قدم ۲- یک تعداد مشخص، $Se \leq N$ ، ژنوم از میان جمعیت حاضر، D_l ، بر اساس یک معیار تعریف شده (معمولاً بهترینها) انتخاب می شوند که با D_l^{Se} مشخص می شود.

قدم ۳- یک مدل احتمالاتی n بعدی که نشان دهنده توزیع n متغیر موجود راه حل می باشد، تخمین زده می شود.

قدم ۴- در پایان یک جمعیت جدید متشکل از N ژنوم جدید با استفاده از تابع توزیع احتمالاتی بدست آمده در مراحل قبل، ایجاد می شود.

قدمهای ۲ تا ۴ الگوریتم فوق تا برقراری شرط خاتمه تکرار می شوند. شرایط خاتمه متفاوتی مانند تولید تعداد معینی نسل، انجام تعداد معینی ارزیابی، یکنواختی در چندین نسل پایانی و ثابت ماندن تقریبی ارزش ژنومها در چند نسل پیاپی را می توان در نظر گرفت. اگر فرض کنید $X_i, i=1, \dots, n$ یک متغیر تصادفی و x_i یک مقدار برای X_i باشد. در این صورت $X = (X_1, \dots, X_n)$ یک بردار تصادفی n بعدی است که می تواند $x = (x_1, \dots, x_n)$ را به عنوان مقدار خود بپذیرد. در صورت پذیرفتن فرض استقلال x_i ها، احتمال یک نمونه x ، برابر حاصلضرب احتمال متغیرهای آن است. به عبارت دیگر داریم:

$$P(X = x) = \prod_{i=1}^n P(X_i = x_i) \quad (23)$$

این مدل، ساده ترین مدل ممکن است و با در نظر گرفتن درصدی از خطا می توان از آن برای حل مسائل بهینه سازی غیرخطی استفاده کرد. در UMDA که یکی از الگوریتمهای اعمال شده در مدل فوق می باشد، توزیع احتمالاتی توأم^۱ ژنوم با استفاده از فراوانی حاشیه ای^{II} هر کدام از متغیرهای ژنوم به صورت زیر بدست می آید.

$$P_l(x_i) = \sum_{j=1}^{Se} \delta_j(X_i = x_i | D_{l-1}^{Se}) / Se \quad (24)$$

به طوریکه اگر متغیر X_i ژنوم l ام مجموعه D_{l-1}^{Se} مقدار x_i داشته باشد $\delta_j(X_i = x_i | D_{l-1}^{Se}) = 1$ می باشد و در غیر این صورت مقدار آن صفر است. ثابت شده است که از نقطه نظر تئوری UMDA رفتاری همانند الگوریتم ژنتیکی ساده خود نشان می دهد [11].

^I Joint Probability Distribution

^{II} Marginal Frequency

۵ - محاسبه گراف میانه عمومی با استفاده از الگوریتم UMDA

در این قسمت، با فرض اینکه مجموعه گراف $S_n = \{G_i = \langle V_i, E_i, \mu_i, \xi_i \rangle \mid 1 \leq i \leq n\}$ و $\bar{G} = \langle \bar{V}, \bar{E}, \bar{\mu}, \bar{\xi} \rangle$ گراف میانه عمومی باشد، مسئله محاسبه \bar{G} را به فرم مناسب برای اعمال الگوریتمهای EDA تبدیل می کنیم. به این جهت در ابتدا به بازنمایی مسئله خواهیم پرداخت و متغیرها، مقادیر آنها، مدل گرافیکی احتمالاتی مورد استفاده و روش ارزیابی را مورد بررسی قرار می دهیم. الگوریتمهای تخمین توزیع همانند الگوریتمهای ژنتیک نیازمند بازنمایی راه حل توسط ساختار مناسبی مانند رشته، ماتریس و یا درخت می باشند. این روش بازنمایی به مسئله مورد بررسی وابسته است. در این بخش ما چند رهیافت برای بازنمایی راه حل در مسئله یافتن گراف میانه عمومی را بیان می کنیم، سپس یکی از آنها را برای ادامه کار خود بر می گزینیم.

ماتریس بییتی: یک ماتریس بییتی $M_{p \times q}$ که در آن $p = \sum_{i=1}^n |V_i|$ و $q = |\bar{V}|$ در نظر بگیرید. عنصر $M_{ii'} = 1$ است، اگر و فقط اگر نود i' ام گراف میانه به نود k ام گراف i ام تصویر شود و داشته باشیم $i = \sum_{l=1}^{l-1} |V_l| + k$ و در سایر موارد $M_{ii'} = 0$ خواهد بود. این ماتریس دارای ساختار بسیار ساده ای است. اما یک ماتریس خلوت¹ می باشد و مقدار زیادی حافظه بدون استفاده، اشغال می کند.

ماتریس صحیح: یک ماتریس $M_{p \times q}$ که در آن $p = n$ و $q = |\bar{V}|$ در نظر بگیرید. عنصر $M_{ii'} = k, 1 \leq k \leq n$ به معنای آن است که نود i' ام گراف میانه به نود k ام گراف i ام مجموعه S تصویر می شود. اگر $M_{ii'} = 0$ آنگاه نود i' ام گراف میانه به هیچ نودی از گراف i ام تصویر نمی شود. این ماتریس از ماتریس بییتی ذکر شده، حافظه کمتری اشغال می کند و ارزیابی آن با سرعت بیشتری صورت می گیرد. در ضمن در این روش بازنمایی، هم گراف میانه عمومی و هم نحوه تصویر شدن آن به گرافهای مجموعه S را در بردارد. از این جهت برای الگوریتم پیشنهادی از این شیوه کد کردن استفاده می کنیم.

آنچه باید به آن توجه داشته باشیم آن است که تمام نمونه هایی که از M ساخته می شوند، مجاز نیستند و در یک ژنوم مجاز نود k ام گراف i ام در سطر i ام حداکثر یک بار تکرار می شود. برای فرار از ایجاد راه حلهای غیر مجاز می توان به چندین روش عمل کرد. ساده ترین راه حل جلوگیری، اعمال تغییرات روی راه حل غیر مجاز به نحوی که محدودیت ارضا شود، می باشد. روشهای غیر مستقیمی نیز وجود دارند، که با گذشت زمان احتمال تولید این راه حلهای غیر مجاز را کاهش می دهند که در ادامه به یکی از آنها خواهیم پرداخت.

روشی که در این جا برای ارزیابی ژنومها استفاده شده است، مشابه روش بکار رفته در [6] می باشد. به این ترتیب که به ازای هر نود \bar{v}_i ($1 \leq i \leq |\bar{V}|$) گراف میانه، گام زیر را اجرا می کنیم:

- برای تمام $l \in L_V$ ، $\bar{\mu}(\bar{v}_i)$ را l قرار می دهیم و مقدار $SOD(\bar{v}_i)$ محاسبه می کنیم:

$$SOD(\bar{v}_i) = \sum_{j \in K_1} c(\bar{\mu}(\bar{v}_i) \rightarrow \mu_j(M_{ij})) + \sum_{j \in K_2} c(\bar{\mu}(\bar{v}_i) \rightarrow \lambda) \quad (25)$$

که مجموعه گرافهایی از S_n که دارای یک نود متناظر با نود i ام گراف میانه عمومی می باشند K_1 را تشکیل می دهند و $K_2 = S_n - K_1$. در این مرحله آن l که $SOD(\bar{v}_i)$ به ازای آن کمترین مقدار را داشته است به عنوان برچسب \bar{v}_i و $SOD(\bar{v}_i)$ مربوط به آن به عنوان سهمیه \bar{v}_i در SOD کل می باشد.

به همین روش به ازای هر (\bar{v}_i, \bar{v}_j) که $i \neq j$ ، گام زیر اجرا می شود:

- برای تمام $l' \in L_E \cup \{\lambda\}$ ، $\bar{\xi}((\bar{v}_i, \bar{v}_j))$ را l' قرار می دهیم و مقدار $SOD((\bar{v}_i, \bar{v}_j))$ را از فرمول (26) با در نظر گرفتن تمام گرافهای مجموعه S بدست می آوریم. در این مرحله آن l' که $SOD((\bar{v}_i, \bar{v}_j))$ به ازای آن کمترین مقدار را داشته است به عنوان برچسب \bar{v}_i و $SOD((\bar{v}_i, \bar{v}_j))$ مربوط به آن به عنوان سهمیه (\bar{v}_i, \bar{v}_j) در SOD کل می باشد.

$$SOD((\bar{v}_i, \bar{v}_j)) = \sum_{k \in l_1} c(\bar{\xi}(\bar{v}_i, \bar{v}_j) \rightarrow l') + \sum_{k \in l_2} c((v_i, v_j) \rightarrow \lambda) \quad (26)$$

به طوری که مجموعه گرافهایی از S_n که دارای یک یال متناظر با یال (\bar{v}_i, \bar{v}_j) گراف میانه عمومی می باشند L_1 را تشکیل می دهند و $L_2 = S_n - L_1$ و در نهایت ارزیابی را با استفاده از فرمول زیر انجام می دهیم:

$$f = -(\sum_{\bar{v}_i \in \bar{V}} SOD(\bar{v}_i) + \sum_{\bar{e} \in \bar{E}} SOD(\bar{e})) \quad (27)$$

برای حذف راه حلهایی که محدودیت مسئله را ارضا نکرده باشند، می توانیم چند استراتژی را اتخاذ کنیم. اولین و ساده ترین استراتژی تغییر مقادیر متغیرهای راه حل غیرمجاز به صورتی که محدودیت ارضا شود، می باشد. در این روش متغیرهایی که در یک سطر ماتریس مقادیر غیر صفر یکسانی را دارند، در نظر گرفته می شود. از میان آنها یکی به تصادف انتخاب شده، سپس در صورت عدم ایجاد مورد غیر مجاز دیگری به آن

¹ Sparse

مقداری غیر صفر داده می شود. در صورت پیدا نکردن یک مقدار مجاز باید به آن مقدار صفر داده شود. این عمل را به طور مداوم تکرار می کنیم تا راه حل به فرم مجاز در بیاید. استراتژی دیگر، تنبیه کردن راه حلهای غیر مجاز است، به این ترتیب که راه حلهای غیر مجاز به گونه ای ارزیابی شوند که شانس برای انتخاب شده در مجموعه D_l^{Se} نداشته باشند.

۶- نتایج شبیه سازی

در این بخش به بررسی نتایج به دست آمده از الگوریتم-۱ (GMG-M) و الگوریتم محاسبه گراف میانه با استفاده از UMDA؛ که آن را تحت نام GMG-UMDA می شناسیم، در مقایسه با نتایج الگوریتم ارائه شده در [6] تحت نام GMG-GA، می پردازیم. داده های آزمایش از میان دو نوع داده گرافی و رشته انتخاب شدند. که در ادامه نتایج هر کدام را شرح خواهیم داد.

همانطور که می دانید، رشته ها را می توان به صورت گراف نمایش داد. به این صورت که هر نماد در رشته توسط یک نود در گراف و توالی بین نمادها به وسیله یالهای جهت دار بین نودهای متناظر با نمادهای متوالی، نشان داده می شود. در جدول (۲) نتایج بدست آمده برای ۳ دسته رشته متفاوت جدول (۱) به نمایش گذاشته شده است. در این آزمایش بازای هر $a \neq b$ $C(\lambda \rightarrow a) = C(a \rightarrow \lambda) = C(a \rightarrow b) = 1$ در نظر گرفته شده است. همانطور که نتایج بدست نشان می دهد، الگوریتم GMG-UMDA در هر سه دسته موفق به محاسبه بهترین رشته میانه شده است. در حالیکه الگوریتم GMG-M در اکثر مواقع نتایجی ما بین رشته میانی عمومی و رشته میانی مجموعه یافته است.

همچنین فرض کنید مجموعه برچسبهای $L_V = \{a, b, c, d, e\}$ و $L_E = \{f\}$ باشند. در ادامه به بررسی نتایج الگوریتم GMG-UMDA، GMG-GA و GMG-M روی گرافهای تولید شده از مجموعه های فوق می پردازیم. برای تولید گرافها از مکانیسم استفاده شده در [13] استفاده کردیم. هزینه های استفاده شده برای عملگرها به صورت زیر می باشند.

$$C(\delta) = \begin{cases} 0.5 & \delta = (\mu(v) \rightarrow l) \text{ and } \mu(v) \neq l \\ 0.5 & \delta = (\xi(v, u) \rightarrow l) \text{ and } \xi(v, u) \neq l \\ 1 & \delta = (v \rightarrow \lambda) \text{ or } \delta = (\lambda \rightarrow v) \text{ or } \delta = ((v, u) \rightarrow \lambda) \text{ or } \delta = (\lambda \rightarrow (u, v)) \\ 0 & \text{otherwise} \end{cases}$$

برای الگوریتم GMG-UMDA و GMG-GA تعداد ژنومهای هر نسل را ۱۰۰۰ (و تعداد ژنومهای انتخاب شده در الگوریتم GMG-UMDA در هر نسل را ۲۵۰ عدد قرار دادیم. همچنین تعداد نودهای تمام گرافهای مجموعه و تعداد نودهای گراف میانه برابر هم فرض شد.

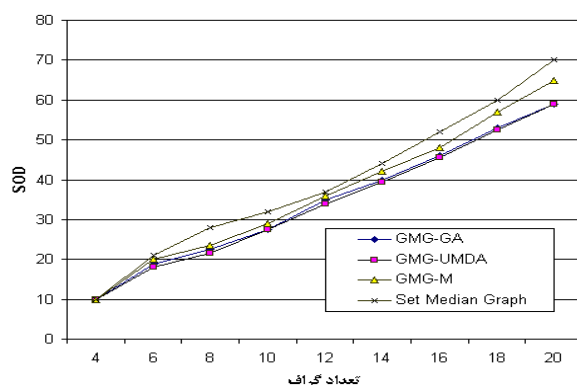
جدول ۱- سه کلاس مختلف از رشته ها

کلاس ۱	کلاس ۲	کلاس ۳
psvholsgy	guohrafhijjq	Ccaph
psychocogr	goraphicsleq	Crcchb
psychokogy	geogrpphical	Gcph
pmvchomog	gvograuhicap	Gbaph
phjcolocq	guographicah	Grbph
psychojozy	geoeraphical	Grbh
psecholody	geogcrgphiccl	Gcabc
psgeochelogy	veograpaical	Grach
psychqfogy	gaographieac	Craph
psycfgholgy	geograhieac	Grah

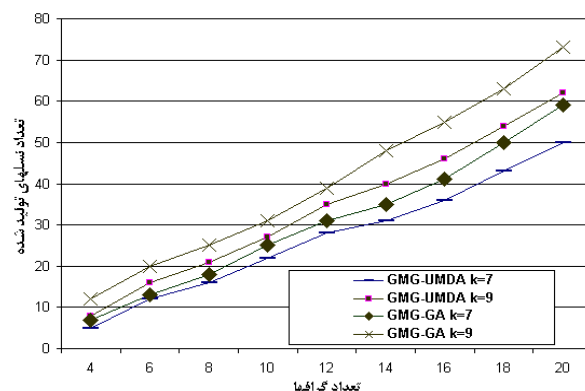
در نمودار (۱) مجموع فواصل گراف میانه بدست آمده برای مجموعه هایی با تعداد اعضای متفاوت ۴، ۶، ۸، ۱۰، ۱۲، ۱۴، ۱۶، ۱۸ و ۲۰ نشان داده شده است. همانطور که دیده می شود گراف میانه بدست آمده از الگوریتم GMG-UMDA نسبت به گراف محاسبه شده توسط GMG-GA اندکی بهبود یافته است. نتایج بدست آمده از الگوریتم GMG-M نیز جالب توجه می باشند. بر اساس مشاهدات می توانیم نتیجه بگیریم به صورت تقریبی میتوان از گراف میانگین بدست آمده توسط الگوریتم (۱) به جای گراف میانه عمومی استفاده کنیم. همچنین نمودار (۲) تعداد نسلهای لازم را برای رسیدن به جواب مناسب در الگوریتم GMG-UMDA و GMG-GA به نمایش می گذارد. بر اساس نتایج بدست آمده الگوریتم GMG-UMDA نسبت به الگوریتم GMG-GA کارایی بهتری را به لحاظ سرعت داراست.

جدول ۲- مقایسه خروجی الگوریتمهای معرفی شده (الگوریتم ۱ و GMG-UMDA) برای ۳ کلاس ارائه شده در جدول (۱)

	کلاس ۱	کلاس ۲	کلاس ۳	SOD1	SOD2	SOD3
رشته میانه مجموعه	psychokogy	georaphical	grach	۳۲	۴۲	۲۱
رشته میانه عمومی GMG-M	psychojogy	gvographical	grah	۳۲	۴۰	۲۱
رشته میانه عمومی GMG-UMDA	psychology	geographical	graph	۲۸	۳۷	۱۹



نمودار ۱- در این نمودار با فرض اینکه تعداد نودهای هر گراف مجموعه و گراف میانه عمومی برابر ۷ است، مجموع فواصل (SOD) به ازای تعداد گرافهای متفاوت به نمایش گذاشته شده است.



نمودار ۲- در این نمودار با فرض اینکه تعداد نودهای هر گراف مجموعه و گراف میانه عمومی برابر k است، تعداد نسلهای تولید شده در الگوریتم GMG-UMDA و GMG-GA برای تعداد گرافهای متفاوت به نمایش گذاشته شده است.

۷- نتیجه گیری

ما در این مقاله، دو الگوریتم تقریبی جدید را برای محاسبه گراف میانه پیشنهاد کردیم. در الگوریتم اول، ابتدا مفهوم میانگین وزن دار دو گراف صفت دار را که اخیراً در [4] معرفی شده است، به مجموعه ای با بیش از دو گراف صفت دار تعمیم داده، سپس گراف میانه عمومی را با محاسبه گراف میانگین برای مجموعه گراف، تخمین زدیم. البته تعمیم میانگین وزن دار برای فضای گرافی می تواند امکان مطالعه آماری بروی فضای گرافی را فراهم سازد. برای مثال با استفاده از آن می توانیم به محاسبه پراش^I در یک مجموعه از ساختارهای نمادین بپردازیم. همچنین با گسترش آن به مفاهیم چولگی^{II} و برجستگی^I بررسی دقیقتری بروی ویژگیهای مجموعه ها در فضای نمادین داشته باشیم. در ادامه مقاله،

^I Variance

^{II} Skewness

الگوریتم دیگری برای محاسبه گراف میانی پیشنهاد دادیم. در این الگوریتم از الگوریتم تخمین توزیع که به منظور رفع مشکلات الگوریتمهای ژنتیکی مطرح شده است، استفاده نمودیم. هدف ما از انتخاب الگوریتم تخمین توزیع برای محاسبه گراف میانه عمومی محک زدن این رهیافت در حل مسائل با پیچیدگی بسیار بالا و ارائه یک الگوریتم ساده و سریع برای محاسبه گراف میانه عمومی بوده است. نتایج بدست آمده از این الگوریتم نشان دهنده قدرت آن در محاسبه گراف میانه با تعداد کمتری ژنوم تولید شده نسبت به الگوریتم GMG-GA [6] می باشد. از رهیافتهایی که می توان در آینده بیشتر بدان پرداخت، بهبود محاسبه گراف میانی عمومی با بررسی سایر الگوریتمهای تکاملی و بخصوص انواع دیگر الگوریتمهای تخمین توزیع و موازی سازی محاسبات، می باشد.

مراجع

- [1] Bunke, H., "Recent Advances in Structural Pattern Recognition with Applications to Visual Form Analysis", Technical Report, Department of Computer Science, University of Bern, Switzerland, 1999.
- [2] Jiang, X., Meunger, A. and Bunke, H., "On Median Graphs: Properties, Algorithms, and Applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 10, pp. 1144-1151, October 2001.
- [3] Wang, S., and Hlaoui, A., "A New Median Graph Algorithm", Technical Report, University of Sherbrooke, Sherbrooke, Canada, 2003.
- [4] Bunke, H., Gunter, S., "A Weighted Means a Pair of Graphs", Technical Report, Department of Computer Science, University of Bern, Switzerland, 1999.
- [5] Gunter, S., Bunke, H., "Self Organizing Map for Clustering in Graph Domain", Pattern Recognition Letter, No. 23, pp. 405-417, 2002.
- [6] Jiang, X., Meunger, A. and Bunke, H., "On Median Graphs: Properties, Algorithms, and Applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 10, pp. 1144-1151, October 2001.
- [7] Singh, M., Chatterjee, A. and Chaudhury, S., "Matching Structural Shape Descriptions using Genetic Algorithms", Pattern Recognition, Vol. 30, No. 9, pp 1451-1461, 1997.
- [8] Khoo, K. G. and Suganthan, P.N., "Structural Pattern Recognition using Genetic Algorithms with Specialized Operators", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 33, No. 1, pp. 156-165, February 2003.
- [9] Larranaga, P., Etxebarria, R., Lozano, J. A., and Pena, J. M., "Optimization by Learning and Simulation of Bayesian and Gaussian Networks", Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of Basque Country, December 1999.
- [10] Pelikan, M., Goldberg, D. E., Lobo, F., "A Survey of Optimization by Building and Using Probabilistic Model", Illinois Genetic Algorithm Report, No. 99018, Illinois University, Illinois, USA, September 1999.
- [11] Mühlenbein, H., and Mahnig, T., "Evolutionary Algorithms: From Recombination to Search Distributions", Theoretical Aspects of Evolutionary Computing, Springer Publication, 2001.

¹ Kurtosis

SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



عضویت در خبرنامه



فیلم های آموزشی

کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی



PROPOSAL
پروپوزال

پروپوزال نویسی و پایان نامه نویسی

دکتره تهرانی

کارگاه آنلاین
پروپوزال نویسی و پایان نامه نویسی



روش تحقیق و مقاله نویسی علوم انسانی

دکتره تهرانی

کارگاه آنلاین
روش تحقیق و مقاله نویسی علوم انسانی



ISI
Scopus

آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو

دکتره تهرانی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو