چاپ شده در مجموعه مقالات: نهمین کنفرانس سالانه انجمن کامپیوتر ایران، دانشگاه صنعتی شریف، ۱۳۸۲، صفحات ۳۴۵ تا ۳۵۲

کلاسه‌بندی فازی بهینه دانشجویان با استفاده از یك تابع فازی در حل مسئله برنامه‌ریزی ژنتیکی دروس هفتگی دانشگاه

محمود امین‌طوسی
کارشناس ارشد کامپیوتر
عضو هیات علمی دانشگاه تربیت معلم سبزوار
m.amintoosi@hsu.ac.ir

هادی صدوقی یزدی
دانشجوی دوره دکتری الکترونیك
دانشگاه تربیت مدرس
sadoghi_y@yahoo.com

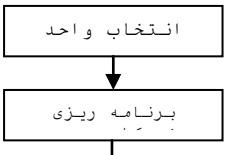
چکیده

تفکیك دانشجویان کلاسهای پرجمعیت (کلاسه‌بندی) یکی از مسائل مهم برنامه‌ریزی هفتگی دروس دانشگاهی می‌باشد. در این مقاله الگوریتم جدیدی در تفکیك دانشجویان ارائه شده، که در آن با یك نمایش جدیدِ بیتی داده‌ها و معیارهای خوشه‌بندی، جداسازی مناسب دانشجویان انجام گردیده است. این تفکیك باعث کاهش میزان تداخل دروس دانشجویان در برنامهٔ هفتگی می‌شود. ابتدا با استفاده از خوشه‌بندی k-میانگین فازی دانشجویان به k دسته تقسیم می‌شوند. سپس با توجه به معیارهای میزان دور بودن مراکز خوشه‌ها، میزان متراکم بودن هر خوشه، میزان هم ورودی بودن دانشجویان هر خوشه و نسبت ابعاد خوشه‌ها و با استفاده از یك تابع فازی، ارزش خوشه‌بندی تعیین می‌شود. با انتخاب ویژگی‌های(دروس) مناسب، بهترین تفکیك دانشجویان بدست می‌آید (بر اساس تابع فازی پیشنهادی و خوشه‌بندی فازی). برای نشان دادن کارائی الگوریتم پیشنهادی، برنامه‌ریزی هفتگی دروس دانشجویان با الگوریتم ژنتیك انجام شده است. با اِعمال الگوریتم پیشنهادی، روی مجموعه‌ای از داده‌های واقعی، نسبت به فرم خوشه بندی بر اساس سال ورود، تعداد تداخلات درسی بطور متوسط ۸% کاهش یافته است. شیوه جدید ارائه شده می‌تواند در بسیاری از الگوریتم‌های مسئله برنامه‌ریزی دروس هفتگی مورد استفاده قرار گیرد.

کلمات‌کلیدی: خوشه‌بندی k-میانگین فازی، تابع ارزش فازی، جداسازی دانشجویان، برنامه‌ریزی دروس، انتخاب ویژگی‌های مناسب، الگوریتم ژنتیك.

۱- مقدمه

مسئله برنامه‌ریزی دروس هفتگی[1] یك گروه یا دانشگاه (TT) عبارت است از انتساب جلسات دروس، به یك سری از بازه‌های زمانی[2] و اتاقها به نحوی که مجموعه‌ای از شرایط مختلف برآورده گردد. برنامه‌ریزی دستی، معمولاً" نیازمند روزها کار بوده و برنامه ریخته شده ممکن است به دلایلی مطلوب نباشد. به عنوان مثال عموما" دانشجویانی پیدا می‌شوند که به علت روی هم افتادن جلسات بعضی از دروس، قادر به انتخاب آنها نمی‌باشند. به علاوه شکل کلی این مسئله جزو مسائل NP-Complete به شمار می رود. بنا به دلایل ذکر شده تلاشهای بسیاری برای خودکارسازی برنامه‌ریزی دروس صورت پذیرفته است. بیشتر تحقیقات انجام شده مبتنی بر الگوریتم های تکاملی [۱-۵]، الگوریتم ژنتیك[۶-۱۰]، گرافها[۱۰-۱۶]، شبیه‌سازی تبریدی[3] [۱۷، ۱۸]، جستجوی تابو [۱۹-۲۱]، شبکه‌های عصبی [۲۲]، شیوه‌های ترکیبی و ابتکاری[4] [۲۳-۲۶] و CLP[5] [۲۷-

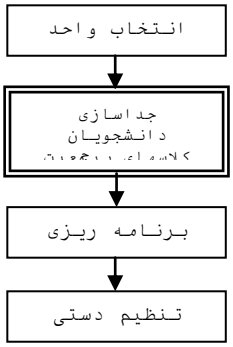


شکل ۱ - روال معمول الگوریتم های برنامه ریزی

[1] Timetabling (TT)– Timetable Construction (TC)
[2] Time Slot
[3] Simulated Annealing
[4] Hybrid and Heuristic Methods
[5] Constraint Logic Programming

۳۱] و اجتماع مورچه‌ها [۳۲] می‌باشد.

روال معمول الگوریتم‌های برنامه ریزی کلاسی به صورت شکل۱ می باشد.



شکل ۲ - روال کلی مسئله برنامه ریزی کلاسی با لحاظ

یکی از مسائل مهم مسئله برنامه‌ریزی دروس، مسئله تفکیك دانشجویان کلاسهای پرجمعیت می‌باشد. تفکیك مناسب دانشجویان یك کلاس پرجمعیت به دو یا چند کلاس کوچکتر در کاهش تعداد تداخلات ناشی از روی هم افتادن دروس مورد انتخاب دانشجویان، موثر است. فقط معدودی از مقالات ارائه شده برای مسئله برنامه ریزی، این مسئله را مورد توجه قرار داده‌اند[۱۶-۱۳، ۱۹و۲۳]. این مسئله نیز همچون مسئله برنامه ریزی هفتگی دروس یك مسئلة NP-Complete می‌باشد[۲۵]. با اضافه کردن این مسئله به مسئلة برنامه‌ریزی کلاسی روال کار به صورت شکل ۲ در خواهد آمد.

در این مقاله الگوریتم جدیدی در تفکیك دانشجویان ارائه شده، که در آن با یك نمایش جدیدِ بیتی داده‌ها و معیارهای خوشه‌بندی، جداسازی مناسب دانشجویان انجام گردیده است. الگوریتم پیشنهادی، در یك سیستم برنامه‌ریزی هفتگی مبتنی بر الگوریتم ژنتیك کارایی خود را نشان داده است.

در بخش بعد تعاریف اولیه مورد نیاز مسئله برنامه‌ریزی دروس و مسئله تفکیك دانشجویان را بیان خواهیم نمود. بخش ۳ به بررسی کارهای انجام شده می‌پردازد. در بخش ٤ نگاهی به خوشه بندی فازی داشته و در بخش پنجم سیستم پیشنهادی را بیان خواهیم نمود. آخرین بخش نیز به بررسی نتایج اجرا روی داده های واقعی اخذ شده از پیش انتخاب ترم دوم سال تحصیلی ۸۲-۸۱ گروه ریاضی دانشگاه تربیت معلم سبزوار اختصاص یافته است.

۲- برنامه ریزی دروس و تفکیك دانشجویان

با افزایش تعداد دانشجویان، تعداد دروس، رشته‌ها و شرایط برنامه‌ریزی، ایجاد یك برنامه هفتگی قابل قبول برای اساتید و دانشجویان، کار مشکلی خواهد شد. فرمولبندی‌های مختلفی برای این مسئله ارائه شده است. فرمولبندی زیر از [۲٦] اخذ شده است:

فرض کنید E رخداد و P بازه زمانی داریم. t_{ip} و c_{ij} را به صورت زیر تعریف می‌کنیم:

t_{ip} : برابر یك است اگر رخداد i در زمان p زمانبندی شده باشد. در غیر این صورت صفر.

c_{ij} : تعداد دانشجویان مشترك دو رخداد i و j.

در مسئله برنامه‌ریزی دروس هر جلسة یك درس، یك رخداد می باشد. در یك برنامه هفتگی قابل قبول هر رخداد باید فقط و فقط یك بار آمده باشد. (معادله ۱). به علاوه اگر دو رخداد در یك دوره زمانی قرار گرفته‌اند، نباید تداخلی با هم داشته باشند (معادله ۲).

معادله ۱
$$\sum_{p=1}^{P} t_{ip} = 1, \forall i \in \{1, \cdots, E\}$$

معادله ۲
$$\sum_{i=1}^{E-1} \sum_{j=i+1}^{E} \sum_{p=1}^{P} t_{ip} t_{jp} c_{ij} = 0$$

معمولاً به جای معادله ۲، شرط مینیمم بودن تعداد کل تداخلات در نظر گرفته می‌شود. مدل فوق شرایط اصلی مسئله را توصیف می‌کند. اما به آسانی قابل توسعه برای لحاظ کردن دیگر شرایط می‌باشد. عموماً این شرایط به دو دسته شرایط سخت و نرم تقسیم می‌شوند[۷]. شرایط سخت آنهائی هستند که در صورت عدم برآورده شدن آنها، برنامه قابل قبول نخواهد بود مانند:

- هر جلسه درسی فقط در یك دوره زمانی قرار گرفته باشد.
- هر استاد در هر دوره زمانی بیشتر از یك درس نداشته باشد.
- در هر اتاق در هر دوره زمانی بیشتر از یك درس نداشته باشیم.
- زمانهای خالی اساتید در نظر گرفته شده باشد.

شرایط نرم آنهایی هستند که رعایت آنها موجب بهتر بودن برنامه خواهد شد، مانند:

- دانشجویان نیز همانند اساتید فضاهای خالی در برنامه خود را نمی‌پسندند.
- دروس به صورت مناسبی در طی هفته پخش شده باشند.
- قرار گیری بعضی دروس در بعد از ظهرها مناسب نیست.
- هر استاد یا دانشجو در هر روز بیشتر از سقف معینی درس نداشته باشد.

به لحاظ شباهت بسیار این مسئله و مسئله برنامه‌ریزی امتحانات یك دانشگاه، بسیاری از محققین روش خود برای مسئله برنامه‌ریزی هفتگی را با مسئله دوم بیان نموده‌اند[۳ و ۸ و۱۰-۱۲].

معمولاً دروسی هستند که دانشجویان زیادی آنها را انتخاب می‌کنند. تفکیك یك چنین درسی به دو یا چند کلاس (بخش) با تعداد دانشجوی کمتر به سه دلیل حائز اهمیت است:

- رعایت ظرفیت معمول کلاسهای دانشگاه.
- رعایت سقف کلاسی قراردادی (مثلا سقف ٦۰ نفر برای دروس عمومی و ٤۰ نفر برای دروس تخصصی).
- تقسیم مناسب یك کلاس به دو یا چند کلاس می‌تواند تعداد تداخلات دروس دانشجویان را کاهش دهد. به عنوان مثال فرض کنید یك گروه از دانشجویان دروس k1 و k2 و گروهی دیگر دروس k1 و k3 را انتخاب نموده و یك جلسه از درس k2 در زمان p و یك جلسه از درس k3 در زمان q قرار داده شده است. در این حالت

درس k1 نمی‌تواند در هیچ یك از زمانهای p یا q قرار گیرد. اما اگر درس k1، بسته به دانشجویان دو گروه، به دو کلاس تقسیم شود، یك کلاس می تواند در زمان p و دیگری در زمان q قرار گیرد. یکسان بودن اندازه هر یك از بخش‌ها نیز قابل توجه می‌باشد[۱۹].

مسئله تفکیك دانشجویان[6] به عنوان یك زیر مسئله از مسئله برنامه ریزی مورد توجه بوده و به صور زیر تعریف می‌شود:

"برای یك برنامه هفتگی داده شده، دانشجویان هر دوره به بخشی از کلاس منتسب شوند که تعداد تداخلات مینیمم گردد"[۳۳].

"پیداکردن بهترین گروه بندی دانشجویان به بخش های کلاسها" [۱۹].

۳- کارهای انجام شده

همانگونه که در مقدمه ذکر شد تحقیقات زیادی برای مسئله برنامه‌ریزی دروس هفتگی انجام شده است. با توجه به آنکه این مقاله روی مسئله تفکیك دانشجویان، به عنوان یك زیر مسئله مهم از TT تکیه دارد، مرور انجام شده در این قسمت، به تحقیقاتی معطوف است که این مسئله را نیز مد نظر قرار داده‌اند.

Selim در [۱٦] ایده جداسازی[۷] رئوس گراف برای کاهش عدد فامی[۸] گراف را معرفی نموده و آنرا برای جداسازی دانشجویان بکار گرفته است. جداسازی یك راس مشابه تفکیك دانشجویان یك درس می‌باشد. با این شیوه Selim توانسته است عدد فامی گراف متناظر با ماتریس تداخل دروس را از ۸ به ۳ کاهش دهد. یعنی برای زمانبندی این دروس فقط به سه بازه زمانی نیاز است.

در[۱۹] از جستجوی تابو برای هر دو مسئله برنامه‌ریزی و تفکیك دانشجویان استفاده شده است. جستجوی تابو در مسئله برنامه‌ریزی دروس نیاز به تعاریف زیر دارد: (الف) یك جواب ممکن، (ب) رابطه همسایگی و روال انتخاب همسایه، (ج) تابع هدفی که باید مینیمم شود، (د) جواب آغازین. در[۱۹]محدودیت‌ها به دسته‌های نرم و سخت تقسیم شده اند. اما یك جواب ممکن بر اساس محدودیت های سخت تعریف نشده است. چرا که در غیر این صورت اولاً مجموعه برنامه‌هایی که در بیشتر شرایط صدق کنند اگر خالی نباشد، ممکن است خیلی کوچك باشد و ثانیاً تضمینی بر پیوسته بودن گراف فضای حالت مسئله وجود نخواهد داشت. به همین دلایل در کار ایشان تعریف یك جواب ممکن مقداری تعدیل شده و برنامه‌هایی که در یك دوره زمانی خاص دارای دروس با استاد یا دانشجویان مشترك باشند، نیز به عنوان جوابی ممکن، پذیرفته می‌شوند. همسایگی N(s) از جواب s شامل تمام برنامه‌هایی است که می توانند از جواب s با انتساب درس c به دوره زمانی t بدست آیند. الگوریتم از بین همسایه های جواب فعلی، فقط بهترینها را انتخاب می‌کند. تابع هدف بکار رفته مجموع وزن دار تعداد تداخلات دانشجویان و اساتید می‌باشد.

برای مسئله تفکیك دانشجویان نیز از جستجوی تابو استفاده شده است. شیوه کلی کار ایشان بر اساس[۲۳] استوار می باشد. ابتدا دانشجویان به بخش‌های کلاسها منتسب شده و یك برنامه هفتگی اولیه تولید می‌شود. به صورت تکراری در هر دور ابتدا بر اساس گروه‌بندی دانشجویان در دور قبل، برنامه هفتگی به منظور کاهش تداخلات اصلاح می شود. سپس با این برنامه جدید گروه‌بندی دانشجویان به منظور کاهش تعداد تداخلات اصلاح می‌شود.

شیوه کلی بکار رفته در[۱٤] به صورت زیر است: الف) جداسازی اولیه دانشجویان[9] ب) در یك حلقه: ابتدا ایجاد یا بهبود برنامه هفتگی (با شیوه ای مبتنی بر شبیه سازی تبریدی[10]) و سپس اصلاح زمانبندی دانشجویان بر اساس این برنامه (مبتنی بر گراف دوبخشی[11]). برخلاف سایر محققینی که انتساب دانشجویان به بخش‌های دروس را فقط به صورت تکراری در کنار ایجاد جدول برنامه هفتگی انجام می‌دهند، Lewandowski جداسازی اولیه را نیز انجام داده است. هدف از جداسازی اولیه انتساب دانشجویان به بخش های دروس (با سقف B نفر) به نحوی است که منجر به یك جدول برنامه هفتگی با کمترین تعداد تداخل شود. اما مشکل اینجاست که در ابتدا جدول برنامه هفتگی در دست نمی‌باشد. ایشان برای جداسازی اولیه دو روش را بررسی نموده است:

راه اول: انتساب دانشجویان با یك ترتیب دلخواه به اولین بخش پر نشده.

راه دوم: مینیمم کردن چگالی گراف بخش ها با خوشه بندی دانشجویانی که برنامه درسی آنها شبیه به هم است. روال اصلی این مرحله اکتشافی بوده و به این ترتیب است که در هر مرحله از یك حلقه، عمومی ترین درس انتخاب شده و حداکثر B دانشجو که این درس و چندین درس مشترك دیگر دارند، در بخش‌های نظیر دروس مربوطه قرار می گیرند.

نتایج ارائه شده نشان می دهد که استفاده از شیوه دوم جداسازی اولیه، نسبت به شیوه اول، به صورت متوسط تعداد تداخل کمتری در نهایت داشته است[۱٤].

از آنجا که در شیوه پیشنهادی در این مقاله از خوشه بندی فازی برای تفکیك دانشجویان استفاده شده است، بخش بعد به مروری بر این موضوع اختصاص داده شده است.

---

[6] Student Sectioning - Grouping Subproblem – Student Scheduling

[۷] Splitting - Subdividing

[۸] Graph Chromatic Number

[9] Initial Sectioning

[10] Simulated Annealing

[11] Bipartite Graph

٤- خوشه‌بندی c میانگین فازی

خوشه‌بندی[12] به معانی مختلف در علوم بکار می‌رود. در بحث طبقه‌بندی بدون راهنما[13] نوعی تحلیل داده[14] به‌شمار می‌رود. کاربردهای متعددی در علوم برای خوشه‌بندی وجود دارد. با معرفی تئوری فازی پروفسور زاده در ۱۹۶۰، خوشه‌بندی فازی نیز شکل گرفت بطوریکه روشهای خوشه بندی فازی متعددی پیشنهاد شد [۳۴و۳۵] .

روشهای خوشه‌بندی فازی به دو دسته تقسیم می‌شوند[۳۶]:

الف- سلسله مراتبی (hierarchical)

ب- بدون سلسله مراتبی (non hierarchical)

خوشه‌بندی c میانگین فازی از نوع بدون سلسله مراتبی است. در نوع سلسله مراتبی تعداد خوشه‌ها ثابت نیستند ولی در نوع c میانگین فازی ثابت است. در c میانگین قطعی[15] ابتدا مراکزی در نظر گرفته شده، سپس نمونه‌هایی که به آن مراکز نزدیکند اختصاص می‌یابند و در مرحلة بعد میانگین جدید محاسبه می‌شود این روند تا عدم تغییر قابل ملاحظة مراکز ادامه می‌یابد.

خوشه بندی c میانگین فازی استاندارد [۳۴و۳۷]

اگر n شیئ دارای p ویژگی باشند، $x_k = (x_{k_1},...,x_{k_p})$ بردار ویژگی شیئ k ام بوده و خواسته باشیم این اشیاء را به c خوشه به صورت مناسب منتسب نماییم، خوشه بندی فازی برای هر شیئ میزان تعلقی به هر خوشه در نظر می گیرد. فرض کنید U یك ماتریس c*n باشد که تعلق هر شیئ را به هر کلاس مشخص می‌کند و $v_i = (v_{i_1},...,v_{i_p})$, $i = 1,...,c$ مراکز کلاسها فرض شود. تابع هدف $j(U,V)$ بر اساس معادله۳ تعریف می‌شود:

$$j(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c} U_{ik}\|x_k - V_i\|^2 \qquad \text{معادله۳}$$

که در آن $\|x_k - V_i\|^2$ نرم اقلیدسی است. و تابع U_i فاصله تمام شیئ‌ها را تا مراکز کلاسها نشان می‌دهد.

$$U_i = \left\{ \{U_{ik}\} \middle| U_{ik} \in [0,i], \sum_{i=1}^{c} U_{ik} = 1 \; for \quad all \quad k \right\}$$

اگر این فاصله حداقل شود، آنگاه خوشه‌های متراکم داریم به نحوی که تعلق آنها به دیگر خوشه‌ها نیز کم است. یا به عبارت بهتر خوشه‌ها از هم دور خواهند بود. بنابراین مینیمم کردن تابع شیئی منجر به ایجاد خوشه‌های متراکم و دور از هم می‌شود.

[12] Clustering
[13] Unsupervised Classification
[14] Data Analysis
[15] Crisp

الگوریتم FC (Fuzzy C-mean) به محاسبه U,V می‌پردازد. این کار با مراحل زیر انجام می‌شود:

۱) تولید U و V بطور اتفاقی

۲) محاسبة U بهینه‌ای که تابع شیئی را حداقل کند، بر اساس معادله زیر:

$$U_{ik} = \left[\sum_{j=1}^{c} \left(\frac{\|x_k - U_i\|^2}{\|x_k - V_i\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \quad ; m \text{ is fuzzifier}$$

۳) محاسبة V بهینه (مراکز جدید)، بنحوی که تابع شیئی را حداقل کند، بر طبق معادله زیر:

$$V_i = \frac{\sum_{k=1}^{n} (U_{ik})^m x_k}{\sum_{k=1}^{n} (U_{ik})^m} \qquad m \geq 2$$

مراحل ۲ و ۳ تا همگرایی تابع شیئی تکرار می‌شوند.

۵- شیوه پیشنهادی

در این کار از اطلاعات مربوط به پیش انتخاب بیشتر از ۲۰۰ دانشجو استفاده شده است. نتایج اجرای الگوریتم خوشه‌بندی سلسله مراتبیMUAS[16] [۳۸] روی این مجموعه داده‌ها برای تفکیك دانشجویان کلاسهای پرجمعیت مطلوب نبوده است. بر طبق این الگوریتم در ابتدا هر دانشجو به عنوان یك خوشه در نظر گرفته می‌شود. سپس تا باقی ماندن فقط دو دسته، در هر مرحله، دو خوشة با بیشترین شباهت با هم ادغام می شوند. معیار شباهت دو دسته، میانگین تعداد دروس مشترك آنها می‌باشد. از آنجا که هر کلاس ممکن است حاوی دانشجویانی باشد که از نظر اشتراك درسی با دیگر دانشجویان، نمونه‌های دور از دیگران باشند این الگوریتم آنها را در یك دسته و تمامی دانشجویان دیگر را در دستة دوم قرار می‌دهد. به عنوان مثال نتایج شبیه‌سازی روی یك کلاس با ۳۶ دانشجو منجر به دو دستة ۱ و ۳۵ تایی شده است. لذا این روش خوشه‌بندی — حداقل با این معیار شباهت- مناسب این مسئله نیست.

در شیوه پیشنهادی از پیش انتخاب دانشجویان، مستقیماً برای خوشه‌بندی استفاده شده است. بردار ویژگی های هر دانشجو یك آرایه منطقی به تعداد کل دروس می باشد. اگر تعداد کل دروس انتخاب شده توسط دانشجویان کلاس مربوطه P بوده و V_i بردار ویژگی دانشجوی i ام باشد، داریم:

$$V_i = (V_{i_1},...,V_{i_P})$$

$$V_{i_j} = 1; \text{if student } i \text{ selects lesson } j \text{ , 0 otherwise}$$

یعنی jامین مولفه بردار ویژگی، در صورتیکه دانشجو درس jام را انتخاب نموده باشد ۱ و در غیر این صورت صفر خواهد بود. به این ترتیب برخلاف خوشه بندی‌های مبتنی بر گراف که فقط شباهت دو دانشجو (تعداد دروس مشترك) ملاك است،

[16] Matrix Updating Algorithmic Scheme

در اینجا نه فقط تعداد دروس مشترك در خوشه‌بندی دانشجویان حائز اهمیت است بلکه تعداد دروسی که هر دو دانشجو انتخاب نکرده‌اند نیز در خوشه‌بندی موثر می باشد.
مثال: جدول زیر انتخاب درس ۵ دانشجو (از بین سه درس A,B,C) و بردارهای ویژگی آنها را نشان می دهد:

جدول۱- نحوه نمایش انتخاب درس دانشجویان

شماره دانشجو	لیست دروس	بردار ویژگی
۱	A,B	110
۲	A,B	110
۳	A,C	101
٤	A,C	101
۵	A,C	101

اگر قرار باشد دانشجویان کلاس درس A به دو دسته تقسیم شوند، دانشجویان ۱و۲ در یك دسته و سه دانشجوی دیگر در دسته دوم قرار خواهند گرفت. نتیجه الگوریتم پیشنهادی نیز همین گونه می‌باشد.

انتخاب ویژگیهای مناسب

همانگونه که در جدول ۱ مشاهده می‌کنید اولین ویژگی در تمام نقاط مشترك است. مسئله انتخاب ویژگیهای مناسب علاوه بر کاهش زمان اجرا، دقت خوشه بندی را افزایش می دهد. در این مسئله، طبیعی به نظر می رسد که دروسی که *اکثریت* یا *اقلیت* دانشجویان کلاس مفروض آنها را انتخاب نموده اند از بردار ویژگی حذف شوند. حذف دروسی که هیچ یك از دانشجویان کلاس انتخاب ننموده اند به عنوان یك مرحله پیش پردازش الگوریتم همیشه انجام می‌شود. مشکلی که می‌ماند تعیین مقادیر درست برای اکثریت و اقلیت است. از آنجا که به سادگی می توان درصد دانشجویانی که هر درس را انتخاب نموده اند مشخص نمود و این مقادیر محدود هستند، با یك جستجوی سراسری می توان آستانه‌های مطلوب را به عنوان مقادیر مناسب *اکثریت* و *اقلیت* برای هر کلاس پیدا نمود شبه کد زیر نحوه جستجو برای پیدا کردن ویژگیهای مناسب و شیوه کلی حل مسئله را نشان می دهد:

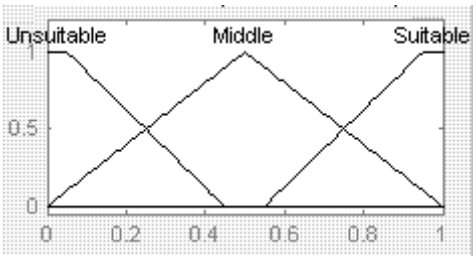
```
AllLessons = Set of all lessons that students of this lesson have took ; //(All Features)
مجموعه دروسی که دانشجویانِ این درس انتخاب نموده‌اند (ویژگیها)//
List1 = Percentage of students that take each lesson, sorted in non increasing order;
درصدِ دانشجویانی که هر درس را انتخاب نموده‌اند، مرتب شده به صورت غیر صعودی//
Thresholds1 = Distinct elements of List1;// عناصر لیست ۱ به صورت غیر تکراری
for i:=1 to length(Thresholds1) do
begin
   MaxSet= Set of Lessons that percentage of students that take them > Thresholds1[i];
   مجموعه دروسی که درصد دانشجویانی که آنها را انتخاب نموده اند بیشتر از آستانه۱ می‌باشد.//
   List2 = Percentage of students that don't take each lesson, sorted in non increasing order;
   درصد دانشجویانی که هر درس را انتخاب ننموده‌اند، مرتب شده به صورت غیر صعودی//
   Thresholds2 = Distinct elements of List2;// عناصر لیست۲ به صورت غیر تکراری
   for j:=1 to length(Thresholds2) do
   begin
      MinSet= Set of Lessons that percentage of students that don't take them > Thresholds2[j];
      مجموعه دروسی که درصد دانشجویانی که آنها را انتخاب ننموده اند بیشتر از آستانه۲ می‌باشد.//
      SelectedFeatures = AllLessons − (MaxSet ∪ MinSet);
      Clusters=FuzzyC-Means(SelectedFeatures);
      if ClusteringEvaluation(Clusters) is better than previous clusters then
      begin
         BestClusters = Clusters;
         BestThreshold1 = Thresholds1[i];
         BestThreshold2 = Thresholds2[j];
      end;  // end if
   end;  //end for j
end;  // end for i
```

در انتها BestThreshold1 مقدار مناسب برای *اکثریت* و BestThreshold2 مقدار مناسب برای *اقلیت* خواهد بود. به این معنی که دروسی که درصد دانشجویانی از این کلاس که آن دروس را انتخاب نموده اند بیشتر از BestThreshold1 باشد یا درصد دانشجویانی که آن

دروس را انتخاب ننموده اند کمتر از BestThreshold2 باشد، از لیست ویژگیها حذف خواهند شد.
سپس عمل خوشه بندی فازی با استفاده از این ویژگیهای انتخاب شده (SelectedFeatures) انجام شده و خوشه بندی ارزیابی می‌شود (ClusteringEvaluation). مقادیر BestThreshold1,2 متناظر با بهترین خوشه بندی می باشند.
یك مسئله مهم نحوه ارزیابی خوشه بندی می باشد. از آنجا که مرحله جداسازی دانشجویان قبل از ایجاد جدول برنامه هفتگی می باشد، هنوز برنامه هفتگی در دسترس نیست که بتوان به صورت قطعی نسبت به ارزش یك خوشه‌بندی نظر داد. به همین لحاظ معیارهای در دسترس زیر پیشنهاد شده و در تابع ارزیاب ملاك عمل قرار گرفته اند:

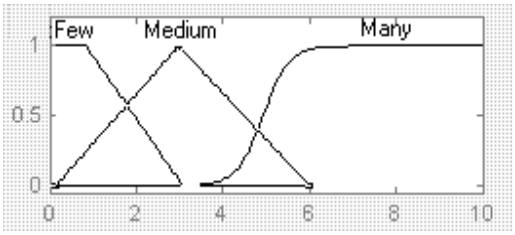
- میزان دور بودن مراکز خوشه ها (ClustersDistance) ،
- میزان متراکم بودن هر خوشه (Density) ،
- نسبت ابعاد خوشه ها (N1PerN2) و
- میزان هم‌ورودی بودن دانشجویان هر خوشه (IDChanging).

هر یك از این پارامترها به عنوان یك متغیر زبانی در یك موتور استنتاج فازی برای ارزیابی خوشه بندی به کار گرفته شده اند. توابع عضویت سه متغیر اول در بازه صفر و یك تعریف شده است. به عنوان مثال، شکل۳ تابع عضویت مربوط به نسبت ابعاد خوشه ها با نام N1PerN2 را نمایش می دهد:



شکل ۳- درجه عضویت نسبت ابعاد خوشه‌ها

میزان متراکم بودن خوشه‌ها برابر با میانگین درصد تراکم دو خوشه در نظر گرفته شده است.
میزان هم ورودی بودن دانشجویان هر خوشه با استفاده از میزان تغییرات سال ورود دانشجویان هر خوشه محاسبه شده است. برای گروهی که دانشجویان دوره کارشناسی دارد به صورت معمول در هر ترم حداکثر شش ورودی فعال دارد. لذا بازه تغییرات سال ورود دانشجویان هر خوشه بین صفر و پنج در نظر گرفته شده است. متغیر زبانی در نظر گرفته شده مجموع تغییرات دو خوشه بوده و لذا بازه آن بین صفر تا ده می باشد. تابع عضویت این متغیر به صورت شکل٤ می باشد:



شکل ٤- درجه عضویت میزان هم ورودی بودن دانشجویان هر خوشه

خروجی با نام Performance دارای ٥ **مقدار** Bad, NotBad, Medium, Good, Excellent می باشد.
تعدادی از قوانین نوشته شده برای موتور استنتاج فازی به قرار زیرند:

If (IDChanging is Few) *and* (Density is High) *and* (ClustersDistance is Good) *and* (N1PerN2 is Suitable) **then** (Performance is Excellent)

If (IDChanging is Few) *and* (Density is High) *and* (ClustersDistance is Good) *and* (N1PerN2 is Middle) **then** (Performance is Good)

If (IDChanging is Few) *and* (Density is High) *and* (ClustersDistance is NotBad) *and* (N1PerN2 is UnSuitable) **then** (Performance is NotBad)

If (Density is Low) *and* (ClustersDistance is NotBad) *and* (N1PerN2 is UnSuitable) **then** (Performance is Bad)

٦- نتایج شبیه‌سازی

در این کار از اطلاعات مربوط به پیش انتخاب دانشجویان در ترم دوم سال تحصیلی ۸۲-۸۱ گروه ریاضی دانشگاه تربیت معلم سبزوار، با مشخصات زیر استفاده شده است:

۱- تعداد دانشجویان: ۲۱۰
۲- تعداد دروس ارائه شده: ۳۸
۳- تعداد جلسات دروس: ٦۲
٤- تعداد اساتید: ۱۳
٥- تعداد کل دروس انتخاب شده توسط دانشجویان: ۱۱٥۷
٦- تعداد بازه‌های زمانی برای برنامه‌ریزی: ۲٥
۷- سقف کلاسی: ٥۰ نفر

برای برنامه‌ریزی کلاسی از الگوریتم ژنتیك استفاده شده است. تابع تطابق الگوریتم ژنتیك برابر با تعداد تداخلات دروس دانشجویان می‌باشد (بدلیل اهمیت این موضوع در الگوریتم پیشنهادی). البته شرایط سخت مسئله مانند عدم روی هم افتادن ساعات درسی اساتید، رعایت ساعات آزاد، سقف کاری اساتید در صبح و دروس تك واحدی لحاظ شده است. مطابق شکل۲ قبل از اجرای الگوریتم ژنتیك با استفاده از الگوریتم پیشنهادی دانشجویان کلاسهای پر جمعیت تفکیك می‌شوند.

در کار ارائه شده در این مقاله تفکیك دانشجویان یك بار قبل از برنامه‌ریزی انجام می‌شود. همانگونه که در ابتدای بخش ذکر شد الگوریتم خوشه بندی MUAS نتایج مطلوبی نداشته است. نتایج شبیه‌سازی نشان می‌دهد الگوریتم پیشنهادی نسبت به فرم معمول انتساب دروس به دانشجویان بر حسب سال ورود، - که معمولاً در برنامه‌ریزی دستی مورد توجه قرار می‌گیرد- بطور متوسط تداخل دروس را به میزان ۸% کاهش داده است.

جدول۲ دروس مورد تفکیك و تعداد دانشجویان هر دسته پس از خوشه بندی را نشان می دهد.

جدول۲- تعداد دانشجویان قبل و بعد از تفکیك

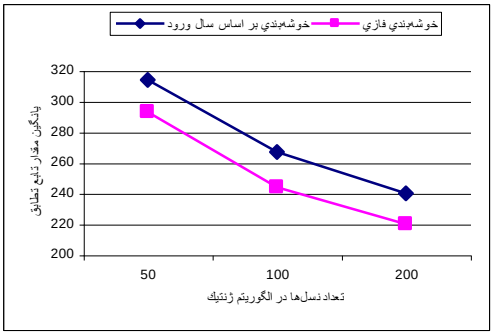
نام درس	تعداد دانشجویان قبل از تفکیك	تعداد دانشجویان هر بخش			
		شیوه پیشنهادی		شیوه مبتنی بر سال ورود	
		بخش اول	بخش دوم	بخش اول	بخش دوم
آموزش ریاضی۱	۷٤	34	40	36	38
جبر۱	٦۷	24	43	14	53
آنالیز عددی۱	٥۲	22	30	20	32

جدول۳ نتایج استفاده از خوشه‌بندی فازی پیشنهادی و خوشه‌بندی بر اساس سال ورود دانشجویان را نشان می دهد. قابل ذکر است که اندازه جمعیت در الگوریتم ژنتیك ۱۰۰ بوده و نتایج هر سطر جدول۳ با ٦۰ بار اجرای الگوریتم ژنتیك برای برنامه‌ریزی دروس بدست آمده اند.

جدول۳- مقایسه روش پیشنهادی و روش مبتنی بر سال ورود

میزان کاهش	خوشه بندی پیشنهادی			بر اساس سال ورود			تعداد نسلها
	Avg	Min	Max	Avg	Min	Max	
%7	293	242	348	314	252	366	٥۰
%9	244	190	294	267	216	338	۱۰۰
%8	220	162	278	240	206	306	۲۰۰
%8	میانگین میزان کاهش تداخلات						

ستون آخر جدول۳ میزان کاهش تعداد تداخلات را در حالت متوسط نشان می دهد. نمودار۱ کمتر بودن متوسط تعداد تداخلات روش پیشنهادی را بهتر نشان می دهد:



نمودار۱: مقایسه متوسط تعداد تداخل ها در روش پیشنهادی و شیوه مبتنی بر سال ورود

یك نمونه از خروجی سیستم آماده شده، در ضمیمه آمده است.

۷- جمع بندی

در این مقاله شیوه جدیدی برای تفکیك دانشجویان کلاسهای پر جمعیت در مسئله برنامه‌ریزی دروس هفتگی دانشگاه ارائه شده است. تفکیك مناسب دانشجویان کلاسها بجز آنکه به لحاظ رعایت سقف کلاسی معمولاً یك ضرورت می‌باشد، می‌تواند تعداد تداخلات درسی دانشجویان در یك برنامه هفتگی را کاهش دهد. بر خلاف شیوه‌های معمول مواجهه با این مسئله که مبتنی بر گراف، و ابتکاری هستند. ساختار پیشنهادی در این مقاله برای نمایش دروس انتخابی دانشجویان، یك آرایه بیتی می‌باشد. برای تفکیك دانشجویان از خوشه‌بندی فازی استفاده شده و برای ارزیابی خوشه‌بندی نیز از یك تابع فازی با چهار معیار میزان دور بودن مراکز خوشه‌ها، میزان متراکم بودن هر خوشه، میزان هم ورودی بودن دانشجویان هر خوشه و نسبت ابعاد خوشه‌ها بهره گرفته شده است. با انتخاب دروس (ویژگیهای) مناسب، بهترین خوشه‌بندی مشخص می‌شود. دروس تفکیك شده با این شیوه خوشه‌بندی و دیگر دروس، به یك الگوریتم ژنتیك جهت برنامه‌ریزی دروس هفتگی داده می‌شود. نتایج بدست آمده نشان می دهد تعداد متوسط تداخلات با شیوه پیشنهادی نسبت به شیوه مبتنی بر سال ورود دانشجویان — که معمولاً در برنامه‌ریزی دستی مورد توجه قرار می‌گیرد- کاهش یافته و لذا برنامه هفتگی مبتنی بر شیوه پیشنهادی ارزش بیشتری خواهد داشت.

تشکر و قدردانی

در خاتمه مؤلفین وظیفه خود می‌دانند که از خانم سمیرا شهرآیینی و آقای حسن فهیمیان دانشجویان گروه ریاضی دانشگاه تربیت معلم سبزوار به خاطر همکاری در پیاده‌سازی و ورود اطلاعات تشکر و قدردانی به عمل آورند.

مراجع

[1] B.Paechter, H. Luchian, M.Petriuc, "Two Solutions to the General Timetable Problem Using Evolutionary Methods", *Proceedings of the First IEEE Conference on Evolutionary Computation,* pp. 300-305, 1994.

[2] D. Corne, P. Ross, H.L. Fang, "Fast Practical Evolutionary Timetabling", *Lecture Notes in Computer Science 865*, pp. 250-263, Springer-Verlag, 1994.

[3] E. K. Burke,J. P. Newall, "A Multi-Stage Evolutionary Algorithm for the Timetable Problem", *IEEE Transactions on Evolutionary* Computation Vol 3.1, pp. 63-74, 1999.

[16]S.M. Selim, "Split Vertices in Vertex Colouring and Their Application in Developing a Solution to the Faculty Timetable Problem", *The Computer Journal*, Vol. 31, No. 1, pp. 76-82, 1988.

[17] D. Abramson, "Constructing School Timetables using Simulated Annealing : Sequential and Parallel Algorithms", *Management Science* 37, pp. 98-113, 1991.

[18] D . Abramson " Simulated Annealing Cooling Schedules for the School Timetabling Problem", *Asia-Pasific Journal of Operational Research* 16, pp. 1-22, 1999.

[19]A. Hertz , "Tabu search for large scale timetabling problems" , *European Journal of Operational Research* 54,pp. 39–47, 1991.

[20] A. Schaerf, "Tabu search techniques for large high-school timetabling problems", *Technical Report CS-R9611*, CWI, Amsterdam, The Netherlands. 1996

[21] A.Schaerf, "Local Search Techniques for Large High School Timetabling Problems", *IEEE Trans. On Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 29, No. 4, pp 367-377, 1999.

[22] H.Mausser, M.J.Magazine, J.B. Moore, "Application of an Annealed Neural Network to a Timetabling Problem", *INFORMS Journal on Computing*, Vol. 8 No. 2, pp. 103-117, 1996.

[23] J. Aubin, , J.A. Ferland, "A large scale timetabling problem", *Computers and Operations Research* 16, pp. 67–77, 1989.

[24] D. Banks, P. Beek, A. Meisels, "A Heuristic Incremental Modeling Aapproach to Course Timetabling", *Proceedings of the Twelfth Canadian Conference on Artificial Intelligence*, pp. 16–29. 1998

[25] R.J.Willemen, *School Timetable Construction, Algorithms and Complexity*, PhD dissertation, Eindhoven : Technische Universiteit Eindhoven, 2002.

[26]J.P. Newall, *Hybrid Methods for Automated Timetabling*, PhD dissertation. Department of Computer Science, University of Nottingham, UK, May 1999.

[27] P. Boizumault, C. Gu'eret, and N. Jussien. "Efficient Labelling and Constraint Relaxation for Solving Timetabling Problems". *Proceedings of the Workshop on Constraint Languages and their use in Problem Modeling, International*

[4] P.Ross, D.Corne, H.L.Fang, "Successful Lecture Timetabling with Evolutionary Algorithms". {peter,dave,hsiaolan}@aisb.ed.ac.uk,Edihnburg

[5] E. K. Burke, J. P. Newall, and R. F. Weare. "Initialization Strategies and Diversity in Evolutionary Timetabling". *Evolutionary Computation*, 6(1), pp. 81–103, 1998.

[6] E .K. Burke, D. G. Elliman, and R. F. Weare. "Specialised Recombinative Operators for Timetabling Problems". In T. C. Fogarty, editor, *Lecture Notes in Computer Science 993* (*AISB Workshop on Evolutionary Computing*), pp 75–85. Springer-Verlag, Berlin, 1995.

[7] W. Erben and Jurgen Keppler. A genetic algorithm solving a weekly course-timetabling problem. In Edmund Burke and Peter Ross, editors, *The Practice and Theory of Automated Timetabling*, pp. 198–211. Springer-Verlag, Berlin, 1996.

[8] W. Erben, "A Grouping Genetic Algorithm for Graph Colouring and Exam Timetabling". *The Practice and Theory of Automated Timetabling III*, pp. 397-421, 2000.

[9] H. Fang, *Genetic Algorithms in Timetabling and Scheduling*, PhD. dissertation, Department of Artificial Intelligence, University of Edinburg, 1994.

[10] E.K.Burke, D.G.Elliman, R.F.Weare, "The Automation of the Timetabling Process in Higher Education", *Journal of Educational Technology Systems*, Vol. 23, No. 4, pp. 257-266, 1995.

[11] E. K. Burke, D. G. Elliman, and R. F. Weare. "Extensions to a University Exam Timetabling System". *Proceedings of the IJCAI-93 workshop on Knowledge-Based Production, Planning, Scheduling and Control*, pp. 42–48, 1993.

[12] E. K. Burke, D. G. Elliman, and R. F. Weare. "A University Timetabling System based on Graph Colouring and Constraint Manipulation". *Journal of Research on Computing in Education*, 27(1), pp. 1–18, 1994.

[13] G.Lewandowski, "Bipartiate Matching for Course Scheduling", cerebro.xu.edu/~lewandow/

[14] G.Lewanowski, "Course Scheduling: Metrics, Models, and Methods", lewan@xavier.xu.edu, 1996.

[15] G.Lewanowski, "Simultaneos Constuction of Student Schedules and Timetable". cerebro.xu.edu/~lewandow/.

Logic Programming Symposium, pp. 116–130, 1994.

[28] H.J. Goltz, G. Küchler, D. Matzke, "Constraint-Based Timetabling for Universities", In *Proceedings INAP'98, 11th International Conference on Applications of Prolog*, pp. 75–80, 1998.

[29] S. Abdennadher, M. Marte, "University Course Timetabling Using Constraint Handling Rules", *Applied Artificial Intelligence*, 14, pp. 311-325, 2000.

[30] M. Henz, J. Wurtz, "Constraint - Based Timetabling – A Case Study", *Applied Artificial Intelligence*, 10, pp. 439- 453, 1996.

[31] H. Rudova, K. Murray, "University Course Timetabling with Soft Constraints", In *PATAT 2002, Proceedings of the 4th international conference on the Practice And Theory of Automated Timetabling*. Belgium, pp. 73-89, 2002.

[32] K.Socha, J. Knowles, M. Sampels, " A MAX-MIN Ant System for University Course Timetabling Problem", *Ant Algorithms Third International Workshop*, Springer-Verlag. pp. 1-13, 2002.

[33] A. Schaerf, *A survey of automated timetabling*, Technical Report CSR9567, CWI, Amsterdam, The Netherlands, 1995, Available at http://www.cwi.nl/ftp/CWIreports/AP.

[34] j. c. Bezdek , *Pattern Recognition with Fuzzy Objective Function Algorithms*, plenum, New York, 1981.

[35] L. Bob Rowski, j. c. Bezdek, "C-means Clustering with the L_1 and L_∞ Norms", *IEEE Trans. on Syst.Man. and Cybern,* Vol. 21, No.3, pp. 545-554, 1991.

[36] S.Miyamoto, "An Overview and New Methods in Fuzzy Clustering", *International Conference on knowledge- Based Intelligent electronic Systems*, pp. 23-40, April 1998.

[37] H. J. Zimmerman, *Fuzzy set theory and it's Applications*, Third Edition, Kluwer Academic publishers, 1996.

[38] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

ضمیمه

جدول زیر یك نمونه از خروجی برنامه می‌باشد. خانه‌های سایه دار، ساعات (فرضی) خالی استاد و دروس مشخص شده، دروس وی را نشان می‌دهند. جلسات هفته در میان نیز با * و ** مشخص شده‌اند. عدد بعد از علامت # کد استاد می‌باشد. دروس سریهای زمانی و فرآیندهای تصادفی سه واحدی در نظر گرفته شده‌اند.

8-10	10-12	14-16	16-18	18-20
فرایندهای تصادفی ۱ #۶ جبر ۳ #۲	انالیز ریاضی ۳ #۱ آمار و احتمال ۱ #۶ ریاضیات گسسته #۴		هندسه دیفرانسیل #۷	انالیزعددی ۲ #۷ آمارو احتمال ۲ #۶
انالیز ریاضی ۱ #۴ ریاضی عمومی۳ #۳ توابع مختلط #۰ جبر ۳ #۲	*طراحی الگوریتمها #۹ **نظریه گراف #۴ جبر ۲ #۲ جبر ۱ #۵	برنامه ریزی آموزش متوس فیزیک پایه ۲ #۱۲	*سریهای زمانی #۶ **فرایندهای تصادفی ۱ # ریاضی عمومی۳ #۳	آموزش ریاضی۱ #۱۱ تحقیق در عملیات ۱ #۸
انالیز ریاضی ۱ #۴ آنالیز ریاضی ۲ #۰ انالیز ریاضی ۳ #۱	*آزمایشگاه فیزیک ۱ #۱۲ **آزمایشگاه فیزیک ۲ #۲	فیزیک پایه ۱ #۱۲ آموزش ریاضی۱ #۱۱	مبانی هندسه #۳ سریهای زمانی #۶ جبرخطی #۲ آنالیزعددی۱ #۷	روانشناسی کودک #۱۱ جبر ۲ #۲ آنالیزعددی۱ #۷
هندسه جبری #۳ نظریه گراف #۴ ساختمان داده ها #۹ تحقیق درعملیات ۲ #۸ جبر ۱ #۵	*هندسه جبری #۳ آنالیزعددی۱ #۷ توابع مختلط #۰	اصول و فنون مشاوره #۱ فیزیک پایه ۲ #۱۲	مبانی هندسه #۳ ریاضیات گسسته #۴ آنالیزعددی۱ #۷	انالیزعددی ۲ #۷ **ساختمان داده ها #۹ نظریه اعداد #۱۰ *جبرخطی #۲
طراحی الگوریتمها #۹ آنالیز ریاضی ۲ #۰ جبر ۱ #۵	آمارو احتمال ۲ #۶ جبر ۱ #۵ تحقیق درعملیات ۲ #۸	فیزیک پایه ۱ #۱۲	تحقیق در عملیات ۱ #۸ آمار و احتمال ۱ #۶ معادلات دیفرانسیل #۷	آموزش ریاضی۲ #۱۱ *هندسه دیفرانسیل #۷ **معادلات دیفرانسیل #۷ نظریه اعداد #۱۰