مدیریت صنعتی
دانشکدهٔ مدیریت دانشگاه تهران
دورهٔ ۵، شمارهٔ ۲
پاییز و زمستان ۱۳۹۲
ص. ۷۰-۴۹

زمان‌بندی درس‌های دانشگاه با به‌کارگیری هیوریستیک فرادست بر مبنای گراف

خداکرم سلیمی‌فرد[۱]، غلامرضا جمالی[۲]، سلمان بابایی‌زاده[۳]

چکیده: زمان‌بندی درس‌های دانشگاه یک مسئلهٔ پیچیدهٔ بهینه‌سازی است. عوامل زیادی مانند گروه‌های آموزشی، استادان، اتاق‌ها و دانشجویان، مسئله را بزرگ و حل آن را دشوار می‌کنند. هر عامل، مجموعه‌ای از محدودیت‌ها را که معمولاً با هم در تضاد هستند، بر فضای حل تحمیل می‌کند. اگر درس‌ها در اتاق‌های متفاوت و در بازه‌های زمانی معین به‌گونه‌ای زمان‌بندی شوند که مجموعه محدودیت‌ها را برآورده کنند، مسئله حل خواهد شد. در این نوشتار برای حل مسئلهٔ زمان‌بندی درس‌های دانشگاه، یک الگوریتم هیوریستیک فرادست بر مبنای گراف ارائه شده است. الگوریتم پیشنهادی یک رویهٔ مدل‌سازی دو لایه است که هیوریستیک فرادست را با روش رنگ‌آمیزی گراف ترکیب می‌کند. لایهٔ بالاتر، یک هیوریستیک مناسب را انتخاب می‌کند که بتواند یک حل‌شدنی خوب را برای مسئلهٔ رنگ‌آمیزی گراف در لایهٔ پایین ارائه دهد. الگوریتم پیشنهادی در حل یک مسئلهٔ واقعی به‌کار رفته است. رویکرد پیشنهادی توانست همهٔ محدودیت‌های نرم و سخت را برآورده کند. بر اساس یافته‌ها می‌توان نتیجه گرفت که رویکرد پیشنهادی یک روش مناسب و کارآی محاسباتی، در یافتن حل مسئلهٔ زمان‌بندی درس‌های دانشگاه است.

واژه‌های کلیدی: ***جست‌وجوی محلی، رنگ‌آمیزی گراف، زمان‌بندی درس‌های دانشگاه، هیوریستیک فرادست.***

۱. استادیار تحقیق در عملیات، گروه مدیریت صنعتی، دانشگاه خلیج فارس، بوشهر، ایران
۲. استادیار تولید و عملیات، گروه مدیریت صنعتی، دانشگاه خلیج فارس، بوشهر، ایران
۳. کارشناس ارشد مدیریت صنعتی، دانشگاه خلیج فارس، بوشهر، ایران

تاریخ دریافت مقاله: ۱۳۹۲/۰۱/۱۹
تاریخ پذیرش نهایی مقاله: ۱۳۹۲/۰٤/۲۹
نویسندهٔ مسئول مقاله: خداکرم سلیمی فرد
E-mail: salimifard@pgu.ac.ir

مقدمه

مسئله‌های زمان‌بندی و برنامه‌ریزی، سازگارسازی و هماهنگ‌کردن مجموعه‌ای از موجودیت‌ها، مانند رخدادها، فعالیت‌ها، افراد، ابزار و دستگاه‌ها، خودروها، مکان‌ها و مانند اینها، در یک الگوی فاصله ـ زمان است. هدف در این‌گونه مسئله‌ها این است که به بهترین شیوه از منابع در دسترس بهره‌برداری شود. همچنین باید محدودیت‌ها و شرایط موجود مورد توجه قرار گیرند و برآورده شوند. در مسئلهٔ زمان‌بندی درس‌های دانشگاه، بایسته است که یگان‌هایی مانند درس‌ها، استادان، کلاس‌ها، روزهای هفته و بازه‌های زمانی برنامه‌ریزی مد نظر قرار گیرند. این یگان‌ها باعث می‌شوند که مدل ریاضی مسئله بسیار بزرگ و پیچیده شود (علیرضایی، خلیلی و منصورزاده، ۱۳۸۵). در بیشتر کاربردها، مدل مسئله چنان بزرگ می‌شود که با ابزارهای موجود حل نمی‌شود. در این هنگام، مجموعه‌ای از الگوریتم‌های ابتکاری برای آن پیشنهاد شده است.

هر سازمان آموزشی در هر سال تحصیلی با مسئله‌های زمان‌بندی گوناگونی چون، زمان‌بندی کلاس‌ها، زمان‌بندی همایش‌ها و زمان‌بندی آزمون‌ها روبه‌روست. شارف (۱۹۹۹)، مسئله‌های زمان‌بندی آموزشی را به سه گروه زمان‌بندی مدرسه، زمان‌بندی درس‌های دانشگاه (درس) و زمان‌بندی آزمون‌ها دسته‌بندی کرده است. افزون‌بر این، در یک پژوهش دیگر (کارتر و لاپرت، ۱۹۹۸)، مسئلهٔ زمان‌بندی آموزشی به پنج گونهٔ تخصیص استاد، زمان‌بندی کلاس ـ استاد، زمان‌بندی درس، زمان‌بندی دانشجویان و تخصیص استاد دسته‌بندی شده است.

یک حل برای مسئلهٔ زمان‌بندی آزمون‌ها، مشخص می‌کند که هر آزمون در کدام روز و در چه زمانی برگزار شود. اگرچه میان زمان‌بندی آزمون‌های دانشگاه و آزمون‌های مدرسه تفاوت زیادی وجود ندارد؛ ولی زمان‌بندی آزمون‌ها ویژگی‌هایی دارد که آن را از زمان‌بندی درس متمایز می‌کند. برای نمونه، هر آزمون تنها مربوط به یک درس است؛ در حالی که یک درس می‌تواند دارای چند نشست هفتگی باشد. همچنین مسئلهٔ زمان‌بندی آزمون‌ها در پی کمینه‌سازی شمار آزمون‌های پی‌درپی دانشجویان است؛ در حالی‌که در زمان‌بندی درس، پی‌درپی بودن نشست‌های درس مطلوب است (بورک و پتروویچ، ۲۰۰۲؛ لوییس، پیچر و روسی‌دوریا، ۲۰۰۷؛ میرحسنی، ۲۰۰۶؛ ون‌دن‌بروک، هارکنس و ووگینگ، ۲۰۰۹).

مسئلهٔ زمان‌بندی درس، دربرگیرندهٔ زمان‌بندی مجموعه‌ای از هم‌نشستی میان استاد و دانشجویان درس، در شمار بازهٔ زمانی است؛ به‌گونه‌ای که محدودیت‌های تعریف‌شده را برآورده کند. یک حل برای مسئلهٔ زمان‌بندی درس، هنگامی شدنی است که هر استاد، دانشجو و کلاس، بیش از یک بار در یک بازهٔ زمانی استفاده نشود. درجه سختی مسئله به شمار دانشجویان، استادان، کلاس‌ها و بازه‌های زمانی برنامه‌ریزی بستگی دارد (میرحسنی، ۲۰۰۶). زمان‌بندی

درس در دانشگاه و مدرسه تفاوت‌های زیادی با یکدیگر دارند. شمار درس‌ها و استادان در دانشگاه بسیار بیشتر از مدرسه است. از این رو، مسئلهٔ زمان‌بندی درس‌های دانشگاه بسیار پیچیده‌تر است (پونگچارون، پروکتت، ینرادی و هیکس، ۲۰۰۸؛ ون‌دن‌بروک، هارکنس و ووگینگ، ۲۰۰۹).

محدودیت‌هایی که در زمان‌بندی درس‌ها باید رعایت شوند را می‌توان به دو دسته سخت و نرم تقسیم کرد. محدودیت سخت، محدودیتی است که برای شدنی‌بودن زمان‌بندی باید برآورده شود. ناهمپوشی درس‌های یک گروه، ناهمپوشی درس‌های یک استاد، شمار و گنجایش کلاس، نمونه‌هایی از محدودیت سخت هستند (بورک و پتروویچ، ۲۰۰۲؛ پونگچارون، پرومتت، ینرادی و هیکس، ۲۰۰۸). محدودیت نرم، گونه‌ای محدودیت است که برآورده‌نشدن آن، بر شدنی‌بودن زمان‌بندی بی‌تأثیر است؛ ولی برآوردن آن به مطلوبیت زمان‌بندی رهنمون می‌شود. به دیگر سخن، کیفیت زمان‌بندی با میزان برآورده‌شدن محدودیت‌های نرم سنجیده می‌شود. محدودیت تخصیص درس، محدودیت زمان کلاس و محدودیت توالی کلاس‌ها (پونگچارون، پرومتت، ینرادی و هیکس، ۲۰۰۸)، نمونه‌هایی از محدودیت نرم هستند. با توجه به نوع سازمان آموزشی (مدرسه یا دانشگاه) و مقررات آموزشی، محدودیت‌ها و نوع آنها می‌تواند متفاوت باشند.

تهیهٔ جدول‌های زمان‌بندی یکی از زمینه‌های مهم پژوهشی است که گروه‌های پژوهشی هوش مصنوعی و پژوهش عملیاتی بر آن تمرکز داشته‌اند (بورک، مک‌کالوم و میزلس، ۲۰۰۷؛ کو و بورک، ۲۰۰۹). روش‌های مختلفی برای حل این مسئله ارائه شده است. یکی از نخستین روش‌های حل این مسئله، مدل‌سازی ریاضی است. در این روش، مسئلهٔ زمان‌بندی به‌صورت یک مدل برنامه‌ریزی خطی یا یک مدل برنامه‌ریزی خطی عدد صحیح فرمول‌بندی می‌شود. در این زمینه پژوهش‌های گسترده‌ای انجام شده است. ازجمله پژوهشگرانی که از برنامه‌ریزی خطی برای حل مسئله زمان‌بندی درس‌ها استفاده کرده‌اند، می‌توان به میرحسنی (۲۰۰۶) و علیرضایی، خلیلی و منصورزاده (۱۳۸۵) اشاره کرد. یکی از سودمندی‌های به‌کارگیری روش‌های دقیق (برنامه‌ریزی ریاضی)، دست‌یافتن به پاسخ بهینهٔ قطعی است. اما به‌کارگیری این روش برای حل یک مسئلهٔ واقعی، نیازمند مدت زمان زیادی، به‌ویژه برای مدل‌سازی و حل است. از این رو، بسیاری از پژوهشگران از مزیت روش‌های دقیق چشم‌پوشی کرده‌اند و به سوی الگوریتم‌های ابتکاری روی آورده‌اند (اکرزلی، ۲۰۰۷).

آبرامسون (۱۹۹۱) و آبرامسون، دنگ و کریشنامورثی (۱۹۹۹) گرم‌کاری شبیه‌سازی‌شده را با موفقیت برای مسائل زمان‌بندی مدرسه به‌کار بردند. هرتز (۱۹۹۱)، یکی از نخستین کسانی بود که از جست‌وجوی ممنوع برای حل مسئلهٔ زمان‌بندی آموزشی استفاده کرد. شارف و شارف

(۱۹۹۵) برای زمان‌بندی جلسه‌های یک دبیرستان از روش جست‌وجوی ممنوع استفاده کردند. راس، کُرون و فانگ (۱۹۹۴) زمان‌بندی گروه‌های درسی دانشگاه را با استفاده از الگوریتم ژنتیک حل کردند. بورک و همکاران (بورک، الیمن و ویر، ۱۹۹۴) یک الگوریتم ژنتیک برای حل مسئلهٔ زمان‌بندی دانشگاه ارائه دادند. آگوستین بلاس و همکاران (بالاس، سانز، گارسیا، فیگوئرا و بلیدو، ۲۰۰۹) نیز یک الگوریتم ژنتیک برای گروه‌بندی دانشجویان برای استفاده از آزمایشگاه پیشنهاد داده‌اند. پونگچارون و دیگران (پونگچارون، پرومتت، ینرادی و هیکس، ۲۰۰۸) از الگوریتم ژنتیک و گرم‌کاری شبیه‌سازی‌شده و نیز جست‌وجوی، تصادفی برای زمان‌بندی درس دانشگاه استفاده کرده‌اند. بورک و دیگران (بورک، نیوال و ویر، ۱۹۹۶) از الگوریتم ممتیک برای زمان‌بندی آموزشی استفاده کرده‌اند. کاراسکو و پاتو (۲۰۰۴) توانستند با به‌کارگیری یک روش ابتکاری برمبنای شبکهٔ عصبی، مسئلهٔ زمان‌بندی کلاس ـ استاد را حل کنند.

بیان مسئله

به‌طور کلی در یک مسئلهٔ زمان‌بندی آموزشی، مجموعه‌ای از رویدادها باید به شمار معینی از بازه‌های زمانی تخصیص داده شوند؛ به شرط آنکه مجموعه‌ای از محدودیت‌ها برآورده شوند. مجموعهٔ $C_g \hat{I} \{C_1, C_2, \dots, C_G\}$ بیانگر درس‌هایی است که در یک نیمسال گروه آموزشی $g\big(g\hat{I}\,\{1, 2, \dots, G\}\big)$ ارائه می‌کند. برای هر درس $c\hat{I}C_g$ پارامتر $C_c^{\min}$ و $C_c^{\max}$، به‌ترتیب بیانگر حد پایین و حد بالای شمار دانشجو در آن درس است. این دو پارامتر را مدیر گروه آموزشی بر اساس ضوابط آموزشی، ترجیح مدرس و گنجایش اتاقی تعیین می‌کند که درس در آن ارائه خواهد شد. همچنین برای هر درس، شمار نشست‌های هفتگی درس نیز تعیین شده است. شمار نشست‌های هفتگی برای درس‌های یک و دو واحدی یک نشست و برای درس‌های سه و چهار واحدی دو نشست است. هر نشست درس‌های دو واحدی در یک بازهٔ زمانی دو ساعتی است. نشست‌های هفتگی یک درس از میان یک مجموعهٔ T از بازه‌های ناهمپوش زمانی برگزیده می‌شود. بازهٔ زمانی هفتگی درس c به‌صورت $T(c)$ تعریف می‌شود. از این رو، اگر $T(c_i) = T\big(c_j\big)$، آنگاه یک دانشجو نمی‌تواند دو درس c_i و c_j را در آن نیمسال بگیرد.

مجموعهٔ $S = \{s | s = 1, \dots, n\}$ بیانگر دانشجویان هر دورهٔ ورودی (سال ورود) گروه آموزشی است. برای هر دانشجوی s، مقدار r_s بیانگر شمار درس‌های درخواستی است. هر دانشجو می‌تواند فهرست $P_s = \{1, \dots, 10\}$ که نشان‌دهندهٔ جایگاه درس در فهرست ترجیح دانشجو است را داشته باشد. برای دانشجویانی که آن نیمسال دانش‌آموخته می‌شوند، شمار درس‌ها در P_s می‌تواند کمتر از ۱۰ باشد. جدول ۱ نمونه‌ای از چند فهرست ترجیحی دانشجویان

را نشان می‌دهد. ستون P_i شناسهٔ درس در جایگاه iام فهرست ترجیحی است. پارامتر c_{sp} بیانگر درس c در جایگاه p فهرست ترجیحی دانشجوی s است.

جدول ۱. نمونه‌ای از فهرست‌های ترجیحی دانشجویان

دانشجو (s)	$P_۱$	$P_۲$	$P_۳$	...	$P_{۱۰}$
۸۹۱۲۳۴۴۵۳	۲۳۰۱۲۴	۲۳۰۲۲۴	۲۳۰۲۱۵	...	۲۳۰۲۴۸
۸۹۱۲۳۴۴۵۴	۲۳۰۲۴۸	۲۳۰۲۵۲	۲۳۰۲۱۹	...	۲۳۰۲۱۵
۸۹۱۲۳۴۴۵۵	۲۳۰۲۴۸	۲۳۰۲۲۴	۲۳۰۲۵۲	...	۲۳۰۲۰۹
۸۹۱۲۳۴۴۵۶	۲۳۰۲۱۹	۲۳۰۲۵۱	۲۳۰۲۳۴	...	۲۳۰۲۵۳

مجموعهٔ I بیانگر همه گروه‌های درسی ارائه‌شده در نیمسال است. هر گروه درسی $i\hat{I}\mathrm{I}$ بیانگر یک درس $c(i)\hat{I}C$ است. بنابراین در هر گروه درسی $c(i)$ شمار دانشجویان باید در بازه C_c^{min} £ $n_{c(i)}$ £ C_c^{max} باشد. زمان‌های هر نشست $i\hat{I}\mathrm{I}$ به‌صورت مجموعه بازه‌های زمانی $T(i)\acute{I}T$ تعریف می‌شود. برنامهٔ آموزشی هفتگی می‌تواند در $R = \{1, 2, \dots, r\}$ اتاق ارائه شود. هر اتاق دارای گنجایش تعریف‌شده‌ای است که با مجموعهٔ $K = \{K_1, K_2, \dots, k_r\}$ تعریف می‌شود. از این رو، برای هر گروه از یک درس که در اتاق $r\acute{I}R$ برنامه‌ریزی می‌شود، باید محدودیت سخت $n_{c(i)}$ £ k_r برآورده شود.

با توجه به آنچه در بالا گفته شد، می‌توان نتیجه گرفت که یک برنامهٔ زمانی باید به‌گونه‌ای باشد که در آن، هر نشست از یک گروه درسی در یک بازهٔ زمانی و یک اتاق تخصیص داده شود؛ به‌گونه‌ای که محدودیت‌های سخت برآورده شود. به دیگر سخن، یک برنامهٔ پذیرفتنی باید محدودیت‌های زیر را برآورده کند:

۱. هر گروه درس باید بر اساس شمار واحد آن، به یک یا چند بازهٔ زمانی تخصیص داده شود؛
۲. هیچ مدرسی نمی‌تواند همزمان به تدریس دو گروه درسی گمارده شود؛
۳. هیچ اتاقی نمی‌تواند همزمان به دو گروه درسی تخصیص داده شود؛
۴. اتاق تخصیص داده‌شده به هر گروه درسی، باید از نوع لازم باشد؛
۵. همهٔ بازه‌های زمانی، مدرسان و اتاق‌های ممنوع و از پیش تخصیص داده‌شده، باید برآورده شوند؛
۶. ساعت‌های درسی یک گروه درسی، نمی‌تواند همپوشی داشته باشند.

پیشینهٔ پژوهش

هیوریستیک فرادست[1]، گونه‌ای هیوریستیک است که برای گزینش هیوریستیک به‌کار می‌رود (بورک و همکاران، ٢٠٠٣) هیوریستیک‌های فرادست به‌جای جست‌وجو در فضای جواب‌های مسئله، روی فضای کاوش هیوریستیک‌ها جست‌وجو می‌کنند. این بدان معناست که هیوریستیک فرادست دارای دو سطح جست‌وجو است. در سطح بالا، از میان شماری هیوریستیک، یکی را برای حل برمی‌گزینند؛ آنگاه در سطح دوم با به‌کارگیری هیوریستیک انتخاب‌شده، مسئله را حل می‌کند. در واقع، هیوریستیک فرادست دارای دو فضای جست‌وجوی الف) فضای جست‌وجوی هیوریستیک سطح بالا و ب) فضای جست‌وجو روی جواب‌های مسئله است. برای مثال در مسئلهٔ زمان‌بندی درس، در هر مرحله از ساخت جواب به‌منظور انتخاب درس برای جای‌گذاری آن در بازهٔ زمانی و اتاق، الگوریتم سطح بالا یکی از هیوریستیک‌های رنگ‌آمیزی را انتخاب می‌کند تا یک زمان‌بندی کامل و شدنی به‌دست آید. به همین روش، در هر تکرار یک ترتیب از هیوریستیک سطح پایین بررسی می‌شود تا ترتیبی از هیوریستیکی انتخاب شود که بهترین زمان‌بندی را ارائه می‌دهد. این بدان معناست که در سطح بالا و برای کار روی فضای جست‌وجوی هیوریستیک‌های سطح پایین، می‌توان از هر هیوریستیک (متاهیوریستیک‌ها، الگوریتم‌های برمبنای جمعیت و مانند اینها) استفاده کرد.

هیوریستیک‌های جست‌وجوشده (در سطح پایین‌تر)، برای حل مسئلهٔ واقعی استفاده می‌شوند. هیوریستیک سطح بالا به‌جای درگیرشدن مستقیم با عناصر اصلی مسئله، در پی انتخاب از میان مجموعه‌ای از هیوریستیک‌های سطح پایین است. هدف از این چارچوب افزایش عمومیت الگوریتم‌های جست‌وجو در حل گسترهٔ وسیعی از مسائل است (کو و بورک، ٢٠٠٩). با تعریف فضای جست‌وجو روی هیوریستیک‌ها به‌جای جواب‌های مسئله واقعی، هیوریستیک فرادست با جزئیات خاص مسئلهٔ واقعی درگیر نخواهد شد و اطلاعات مسئلهٔ اصلی به هیوریستیک‌های سطح پایین واگذار می‌شود. بیان فضای جست‌وجو برای هیوریستیک سطح بالا معمولاً ساده است و پیچیدگی بیان جواب‌های مسئلهٔ واقعی را ندارد. نقش جست‌وجوی سطح بالا، اداره‌کردن مجموعه‌ای از هیوریستیک‌های سطح پایین است که می‌تواند برای گستره‌ای از مسائل به‌کار برده شود.

ایدهٔ هیوریستیک فرادست در سال ١٩٦٠ ارائه شد (بورک و همکاران، ٢٠٠٣)؛ اگرچه در آن زمان این‌گونه نام‌گذاری نشده بود. هیوریستیک فرادست با به‌کارگیری شماری از الگوریتم‌ها بیان می‌شود با وجود این، تفاوت بنیادینی با بیشتر رویکردهای هیوریستیک دارد. برای مثال، در

1.Hyper-heuristic

زمان‌بندی پروژه با منابع محدود، جایگشت (ترتیب) کارها معمولاً با استفاده از یک استراتژی مشخص به‌دست می‌آیند و سپس کارها یک‌به‌یک با استفاده از این ترتیب خاص، زمان‌بندی می‌شوند (براکر و کناست، ۲۰۰۶). در هیوریستیک فرادست، هدف جست‌وجوی مجموعه‌ای از استراتژی‌ها (در سطح پایین) است که کارها را برای ساخت جواب‌های با کیفیت بهتر در طول زمان‌بندی جایگزین می‌کنند.

بورک و همکاران (بورک، کندال و سوبیگا، ۲۰۰۳) برای حل مسئلهٔ زمان‌بندی درس و پرستار، جست‌وجوی ممنوع را هیوریستیک سطح بالا روی مجموعه‌ای از استراتژی‌های حرکت شمرده و از آن استفاده کرده‌اند. در پژوهش (اکرزلی، ۲۰۰۷) برای تعیین اندازهٔ کشتی در یک مسئلهٔ حمل‌ونقل، یک رویکرد گرم‌کاری شبیه‌سازی‌شده، هیوریستیک سطح بالا فرض‌شده و از آن استفاده شده است. در مطالعهٔ (گآو، رتادیلوک، کوآن، ۲۰۰۵) برای مسئلهٔ زمان‌بندی آزمون‌ها، یک تابع انتخاب توزیع‌شده[1] که هیوریستیک سطح بالا فرض شده بود، به‌کار گرفته شد.

دستهٔ دیگری از کاربردهای هیوریستیک فرادست، هیوریستیک‌های ترتیبی را هیوریستیک‌های سطح پایین فرض کرده‌اند. در پژوهش بیلگین، اوزجان و کورماز، ۲۰۰۷) برای مسائل زمان‌بندی آزمون‌ها و توابع بهینه‌سازی، انتخاب هیوریستیک و معیار پذیرش حرکت در هیوریستیک سطح بالا بررسی شده است. در مطالعهٔ (بورک، پتروویچ و کو، ۲۰۰۶) برای مرتب‌کردن رویدادها، هیوریستیک‌های رنگ‌آمیزی گراف بر پایهٔ استدلال مبتنی بر محدودیت[2] به‌کار گرفته شد. راس و همکاران (راس، مارین ـ بلازکوئز و هارت، ۲۰۰۴) هیوریستیک‌های انتخاب رویداد و بازهٔ زمانی را با الگوریتم ژنتیک انجام دادند و مسئلهٔ زمان‌بندی درس و آزمون را با این رویکرد حل کردند. تراشیمامارین و همکاران (مارین، راس و رندون، ۱۹۹۹) زمان‌بندی آزمون‌های استراتژی‌های برآورده‌سازی محدودیت را با الگوریتم‌های تکاملی انتخاب، انجام دادند.

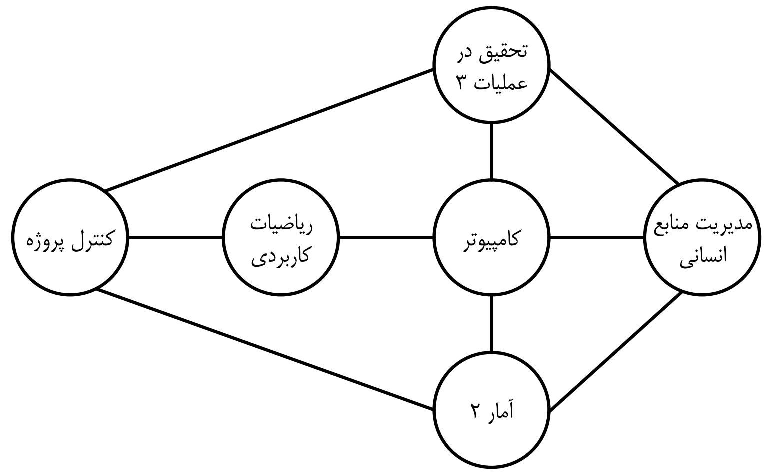
روش پژوهش

در این پژوهش برای حل مسئلهٔ زمان‌بندی درس‌های دانشگاه با به‌کارگیری رنگ‌آمیزی گراف، متغیر تصمیم به‌شیوهٔ زیر تعریف شده است:

$$x_{ijkl} = \begin{cases} 1 & \text{درس } i \text{ با استاد } j \text{ در روز } k \text{ و در بازه زمانی } l \text{ برگزار شود.} \\ 0 & \text{وگرنه} \end{cases}$$ رابطهٔ ۱)

1. Distributed selection function
2. Constraint-based reasoning

با به‌کارگیری تعریف بالا و بر اساس نظریه گراف، مسئله به‌شیوهٔ زیر تعریف می‌شـود. گـراف $G = (V, E)$ در بر گیرندهٔ $V = \{v_1, v_2, \dots, v_n\}$ گره و $E = \{e_1, e_2, \dots, e_m\}$ یـال اسـت. هر یال گراف دارای دو گره $v_j \hat{I} V$ و v_i است. هر گره $v_i \hat{I} V$ بیانگر یـک متغیـر x_{ijk} اسـت. یال $e_j \hat{I} E$ که دو گره v_i و v_l را به هم پیوند می‌دهـد، نشـان‌دهنـدهٔ تضـاد میـان متغیرهـای تصمیم متناظر است.



شکل ۱. گراف زمان‌بندی درس‌ها

در شکل یال میان گره‌های کامپیوتر و آمار ۲ بدین معناست که این دو درس با هم در تضاد هستند و نمی‌توانند در یک بازهٔ زمانی همسان زمان‌بنـدی شـوند. از آنجـا کـه میـان کـامپیوتر و کنترل پروژه، یال نیست، این دو درس را می‌توان در بازهٔ زمانی همسان زمان‌بندی کرد.

با به‌کارگیری متغیر تعریف‌شدهٔ بالا، مدل ریاضی مسئله نیازمند تعریف محدودیت‌هایی اسـت که بتواند بیانگر شرایط حاکم بر مسئله باشد. در این پژوهش محدودیت‌های مسئله بـه دو دسـته محدودیت‌های سخت (که برآورده‌نشدن هر یک از آنها بیانگر ناشدنی (ناموجه) بودن حل اسـت) و محدودیت‌های نرم (که در یک حل‌شدنی می‌توانند برآورده نشوند، ولی برآورده‌شدن آنها بیانگر بهتر بودن حل است)، گروه‌بندی می‌شوند. در ادامه محـدودیت‌هـای سـخت مسـئله و چگـونگی مدل‌سازی آنها آورده می‌شود:

۱. نبود همپوشی در درس‌های دانشجو: درس‌های ارائه‌شده برای دانشجویان هر ورودیِ (سال ورود) هر گروه آموزشی، نباید دارای همپوشی زمانی باشند. این محدودیت برای هر درس i، و برای هر بازه l در هر روز k به‌شیوهٔ رابطهٔ ۲ مدل‌سازی می‌شود.

$$\sum_{i} x_{ijkl} \leq 1 \qquad \forall k,l$$ رابطهٔ ۲)

۲. برگزاری درس در بازهٔ پیشنهادی استاد: برنامهٔ درسی هر استاد به‌گونه‌ای زمان‌بندی شود که تدریس یک درس برای هر استاد در یکی از بازه‌های زمانی وی باشد. با معرفی متغیر s_i که بیانگر شمار نشست‌های هفتگی هر درس i است، رابطهٔ ۳ شیوهٔ مدل‌سازی این محدودیت را نشان می‌دهد.

$$\sum_{k,l} x_{ijkl} = s_i \qquad \forall i$$ رابطهٔ ۳)

۳. نبود همپوشی در برنامهٔ استاد: برنامهٔ درسی باید به‌گونه‌ای زمان‌بندی شود که برای هیچ استاد j، برنامهٔ درسی وی در بازه‌های زمانی l هر روز k دارای همپوشی نباشد. این محدودیت به‌شیوهٔ رابطهٔ ۴ مدل‌سازی می‌شود.

$$\sum_{i} x_{ijkl} \leq 1 \qquad \forall j,k,l$$ رابطهٔ ۴)

۴. شمار اتاق‌ها: در هر بازهٔ زمانی نباید بیش از یک کلاس درس به هر اتاق تخصیص داده شود. با معرفی متغیر N که بیانگر شما اتاق‌های در دسترس در هر بازهٔ زمانی است، رابطهٔ ۵ شیوهٔ مدل‌سازی این محدودیت را برای هر بازهٔ l در هر روز k نشان می‌دهد.

$$\sum_{i} x_{ijkl} \leq N \qquad \forall k,l$$ رابطهٔ ۵)

۵. گنجایش اتاق: شمار دانشجو در هر درس نباید بیش از گنجایش اتاقی باشد که درس به آن تخصیص داده شده است. در این پژوهش، اتاق‌های در دسترس به دو گروه اتاق‌های با گنجایش ۳۰ نفر (Nt) و اتاق‌های با گنجایش کمتر از ۳۰ نفر دسته‌بندی شده‌اند. رابطهٔ ۶ مدل ریاضی این محدودیت را نشان می‌دهد.

$$\sum_{i \mid p_i > 30} x_{ijkl} \leq N_t \qquad \forall k,l$$ رابطهٔ ۶)

۶. یک نشست برای هر درس در یک روز: هر درس i می‌تواند بـا توجـه بـه ارزش آن (واحـد درس) در یک یا دو نشست هفتگی برگزار شود. درس‌هـای یـک و دو واحـدی دارای یـک نشست هفتگی و درس‌های سـه و چهـار واحـدی دارای دو نشسـت هفتگـی هسـتند. ایـن محدودیت بیانگر حداکثر یک نشست در همهٔ بازه‌های هر روز k برای درس i است. رابطهٔ ۷ مدل ریاضی این محدودیت را نشان می‌دهد.

$$\sum_{l} x_{ijkl} \le 1 \qquad \forall i,k$$ رابطهٔ ۷)

افزون‌بر محدودیت‌های سخت مدل‌شده در رابطه‌های ۲ تا ۷، در این پژوهش محدودیت‌های نرم زیر نیز به‌شرح زیر در نظر گرفته شده است:

۱. دست‌کم دو و حداکثر سه کلاس برای دانشجو: برای دانشجو خواستنی این است که شـمار کلاس‌ها در یک روز نه خیلی کم و نه آنچنان زیاد باشد که خسته‌کننده شود. از ایـن رو در این پژوهش شمار خواستنی درس‌ها برای هر دانشجو در یک روز دست‌کم دو و حداکثر سه کلاس تعریف شده است.

۲. پیاپی بودن کلاس‌های دانشجو: هر دانشجو می‌خواهد که تا حد ممکن کلاس‌های درس در هر روز در بازه‌های زمانی پی‌درپی ارائه شود.

دو محدودیت نرم یادشدهٔ بالا با رابطهٔ ۸ مدل شده است. در این مدل، ضرایب y_i به‌گونـه‌ای انتخاب شده است که محدودیت‌ها به بهترین شیوهٔ ممکن برآورده شوند.

$$\begin{aligned}
&\sum_{i,l} x_{ijkl} = y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 \qquad \forall k\,, \\
&\sum_{i=1}^{5} y_i \le 1,\ \sum_{i,l=1,2} x_{ijkl} = y_6 + 2y_7,\ \sum y_6 + y_7 \le 1 \\
&\sum_{i,l=2,3} x_{ijkl} = y_8 + 2y_9,\ \ \sum y_8 + y_9 \le 1, \\
&\sum_{i,l=3,4} x_{ijkl} = y_{10} + 2y_{11} \quad , \quad \sum y_{10} + y_{11} \le 1 \\
&\sum_{i,l=4,5} x_{ijkl} = y_{12} + 2y_{13} \quad , \quad \sum y_{12} + y_{13} \le 1 \qquad y_i \in \{0,1\}
\end{aligned}$$ رابطهٔ ۸)

۳. نبود دو نشست درس در یک روز: اگر یک درس دارای دو نشست هفتگی باشد، دسـت‌کـم یک روز میان نشست‌های هفتگی آن درس فاصله باشد. رابطهٔ ۹ چگونگی مدل‌سـازی ایـن محدودیت نرم را نشان می‌دهد.

$$\sum_{k,k+1} \sum_{l} x_{ijkl} = y_{14} + 2y_{15}$$
$$\sum y_{14} + y_{15} \le 1 \qquad y_i \in \{0,1\}$$
رابطهٔ ۹)

۴. نبود کلاس در آخرین بازهٔ روز: دانشجو ترجیح می‌دهد که در آخرین بازهٔ زمـانی روز (دامنـهٔ زمانی ۱۸:۰۰ تا ۲۰:۰۰) کلاس نداشته باشد. شیوهٔ مدل‌سازی این محدودیت نـرم در رابطـهٔ ۱۰ نشان داده شده است.

$$\sum_{i,k,l=5} x_{ijkl} = M$$
رابطهٔ ۱۰)

در رابطهٔ ۱۰، M جریمـه‌ای اسـت کـه از تخصـیص کـلاس در بـازهٔ زمـانی پـنجم ($l=5$) پیشگیری می‌کند.

۵. پیوستگی برنامهٔ روزانهٔ استاد: کلاس‌های درس تخصیص داده‌شده برای یک استاد در یـک روز تا جایی که ممکن است در بازه‌های زمانی پیـاپی باشـد. رابطـهٔ ۱۱ شـیوهٔ مـدل‌سـازی این‌گونه محدودیت نرم را نشان می‌دهد.

$$\forall j,k:$$
$$\sum_{i,l=1,2} x_{ijkl} = y_{16} + 2y_{17} \quad , \sum y_{16} + y_{17} \le 1$$
$$\sum_{i,l=2,3} x_{ijkl} = y_{18} + 2y_{19} \quad , \sum y_{18} + y_{19} \le 1$$
$$\sum_{i,l=3,4} x_{ijkl} = y_{20} + 2y_{21} \quad , \sum y_{20} + y_{21} \le 1$$
$$\sum_{i,l=4,5} x_{ijkl} = y_{22} + 2y_{23} \quad , \sum y_{22} + y_{23} \le 1 \qquad y_i \in \{0,1\}$$
رابطهٔ ۱۱)

در این پژوهش تابع هدف از گونهٔ بیشینه‌سازی بوده و دربرگیرندهٔ محدودیت‌های نرم اسـت. این بدان معناست که تابع هدف، بیانگر میزان برآورده‌شدن محدودیت‌های نرم یا به دیگر سخن، بیانگر کیفیت حل است. با برآورده‌نشـدن هـر محـدودیت نـرم، تـابع هـدف بـه‌میـزان از پـیش تعریف‌شده‌ای جایزه یا جریمه می‌گیرد. تابع هـدف اجزایـی دارد کـه هـر یـک بـر اسـاس یـک محدودیت نرم تعریف می‌شود.

– اگر برای دانشجویان ورودی یک دوره از یک گروه آموزشی، تنها یک نشست در یـک روز، یا بیش از سه نشست در روز تخصیص داده شود، جریمه‌ای به‌میزان ۲ بـرای هـر نشست در نظر گرفته شود.

- اگر برای دانشجویان ورودی یک دوره از یک گروه آموزشی، دو یا سه نشست در روز برنامه‌ریزی شود، ۲ واحد جایزه در نظر گرفته شود.
- بر اساس محدودیت نرم شماره ۲، اگر در یک روز دو یا چند نشست غیر پیاپی برای دانشجویان برنامه‌ریزی شود، ۲ واحد جریمه لحاظ شود.
- اگر دو نشست هفتگی یک درس در دو روز پیاپی برنامه‌ریزی شود، تابع هدف یک واحد جریمه می‌شود.
- اگر یک نشست از یک درس در آخرین بازهٔ روز زمان‌بندی شود، تابع هدف یک واحد جریمه می‌شود.
- اگر برای یک استاد در یک روز تنها یک کلاس زمان‌بندی شود، تابع هدف ۲ واحد جریمه می‌شود. اگر یک استاد در یک روز تا چهار کلاس در بازه‌های زمانی پیاپی داشته باشد، تابع هدف ۲ واحد جایزه می‌گیرد و در غیر این صورت، ۲ واحد جریمه می‌شود.

بر اساس آنچه گفته شد، تابع هدف مسئله بر اساس رابطهٔ ۱۲ مدل‌سازی می‌شود. اگر یک حل بتواند این تابع هدف را بیشینه کند، حل یادشده محدودیت‌های نرم را به بهترین شیوه برآورده کرده است.

$$\begin{aligned} Max\,Z = &-2y_1 - y_2 - 2y_3 - 6y_4 - 10y_5 + 2y_7 + 2y_9 \\ &+ 2y_{11} + 2y_{13} - 2y_{15} - M + 2y_{17} + 2y_{19} + 2y_{21} + 2y_{23} \end{aligned} \qquad \text{(۱۲ رابطهٔ)}$$

برای حل مسئلهٔ زمان‌بندی درس‌های دانشگاه، مدل ریاضی بالا به یک مسئلهٔ رنگ آمیزی گراف نگاشت می‌شود.

هیوریستیک‌های رنگ‌آمیزی گراف

یک گراف $G = (V, E)$ مجموعه‌ای از گره‌ها ($V = \{v_1, v_2, \dots, v_n\}$) و یال‌های بدون جهت ($E = \{e_1, e_2, \dots, e_m\}$) است که گره‌ها را به هم پیوند می‌دهد. دو گره v_i و v_j را همجوار گویند، اگر که یک یال این دور گره را به هم پیوند دهد.

رنگ‌آمیزی گراف یک

هیوریستیک‌های رنگ‌آمیزی گراف، هیوریستیک‌های سازنده‌ای هستند که رویدادها (مانند درس یا آزمون در مسئلهٔ زمان‌بندی) را براساس معیار سختی مرتب می‌کنند. سپس رویدادهای مرتب‌شده یک‌به‌یک با شروع از دشوارترین رویداد، تخصیص داده می‌شوند. فرض اساسی این

است که رویدادهای سخت‌تر باید زودتر زمان‌بندی شوند تا در مرحله‌های بعدی، مسئله پیچیده‌تر نشود. فهرستی از هیوریستیک‌های رنگ‌آمیزی گراف که به‌شیوه‌ای گسترده در زمان‌بندی استفاده شده‌اند، در جدول ۲ ارائه شده است.

جدول ۲. استراتژی‌های ارزیابی سختی برمبنای هیوریستیک‌های گراف برای مسائل زمان‌بندی

نام هیوریستیک	استراتژی
بزرگترین درجه	بر اساس تعداد تعارضی که یک رویداد با بقیه رویدادها دارد.
درجهٔ رنگ	بر اساس تعداد تعارضی که با رویدادهای زمان‌بندی‌شده دارد.
درجهٔ اشباع	براساس تعداد بازه‌های زمانی مجاز که برای رویداد در دسترس است.
بیشترین ثبت‌نام‌شده	رویدادها براساس تعداد متقاضی برای هر رویداد مرتب می‌شود.
بزرگترین درجهٔ وزنی	همانند بالایی اما وزن‌گذاری تعارض‌ها با شمار دانشجویان مشترک.

برای مثال اگر از هیوریستیک درجهٔ اشباع برای زمان‌بندی استفاده شود، رویدادها بر اساس تعداد بازهٔ زمانی معتبر که می‌توانند در آن قرار بگیرند، مرتب می‌شوند (رویدادی که کمترین بازه زمانی را دارد زودتر زمان‌بندی می‌شود).

در پژوهش‌های پیشین (بورک، کینگستون و دورا، ۲۰۰۴) نشان داده شده است که هیوریستیک‌های رنگ‌آمیزی گراف را می‌توان به خوبی برای روش‌های ساخت جواب اولیهٔ متاهیوریستیک‌ها به‌کار برد یا از ترکیب هوشمند آنها برای ساخت زمان‌بندی‌های پیچیده و مسائل بهینه‌سازی استفاده کرد آسومنی، بورک و گاریبالدی، ۲۰۰۵؛ بورک، درور، پتروویچ و کو، ۲۰۰۵، بورک، پتروویچ و کو، ۲۰۰۶). این موضوع به پژوهشگران انگیزه می‌دهد تا در استفاده از هیوریستیک فرادست برای ساخت جدول زمان‌بندی، از هیوریستیک‌های رنگ‌آمیزی گراف برای هیوریستیک‌های سطح پایین بهره‌گیری کنند. در این مقاله، به‌دلیل ماهیت مسئله و همچنین برای دست‌یابی به‌اجرای بهتر، از سه هیوریستیک نخست در جدول ۲ برای انتخاب رویداد و زمان‌بندی آن استفاده شده است.

رویکرد هیوریستیک فرادست برمبنای گراف

در هیوریستیک فرادست برمبنای گراف که بورک و همکاران (بورک، مک‌کالوم، میزلس، ۲۰۰۷) ارائه کرده‌اند، توالی هیوریستیک‌های رنگ‌آمیزی گراف با هیوریستیک جست‌وجوی ممنوع بررسی می‌شود و در ساخت زمان‌بندی به‌کار می‌رود. در بیشتر کارهای انجام‌شده روی هیوریستیک‌های کلاسیک (مانند هیوریستیک‌های رنگ‌آمیزی گراف) با تخصیص رویدادهای

منظم و با ترتیب خاص، جواب به‌دست می‌آید. اما این رویکردها در بیشتر موارد به جواب ناشدنی دست می‌یابند (کو و بورک، ۲۰۰۹). رویکرد هیوریستیک فرادست برمبنای گراف، روی ترتیب‌های متفاوت هیوریستیک‌های رنگ‌آمیزی گراف جست‌وجو می‌کند تا هنگام ساخت جواب، رویدادها را با استفاده از استراتژی‌های متفاوتی تخصیص دهد. نقش هیوریستیک سطح بالا، یافتن بهترین ترتیب هیوریستیک‌ها برای ساخت جواب زمان‌بندی است. کیفیت هر ترتیب هیوریستیک با کیفیت زمان‌بندی متناظر با آن سنجیده می‌شود. بنابراین هیوریستیک سطح بالا درپی یافتن دنباله‌ای از هیوریستیک‌ها است که یک جدول زمان‌بندی با بهترین میزان تابع هدف را تولید کند.

رویکرد هیوریستیک فرادست برمبنای گراف، دو فضای جست‌وجوی هیوریستیک سطح بالا و فضای جست‌وجوی جواب‌های واقعی را مشخص می‌کند. هر توالی هیوریستیک (که متناظر با یک نقطه در فضای جست‌وجوی هیوریستیک سطح بالا است) در فضای جست‌وجو، یک جواب کامل را می‌سازد که متناظر با نقطه‌ای در فضای جواب مسئلهٔ واقعی است. کیفیت جواب در فضای جواب واقعی، به‌منزلهٔ تابع هدف برای هیوریستیک سطح بالا استفاده می‌شود. جواب‌هایی که فهرست‌های همانند (فهرست‌های همسایه) ساخته‌اند، در فضای جست‌وجو ممکن است از نظر جواب مسئلهٔ واقعی با هم بسیار متفاوت باشند. این بدان معناست که هیوریستیک فرادست بر مبنای گراف (GbHH)، می‌تواند جواب‌های موجود در نقاط مختلف فضای جواب را با حرکت‌های همسایگی در هیوریستیک سطح بالا پوشش دهد و بررسی کند.

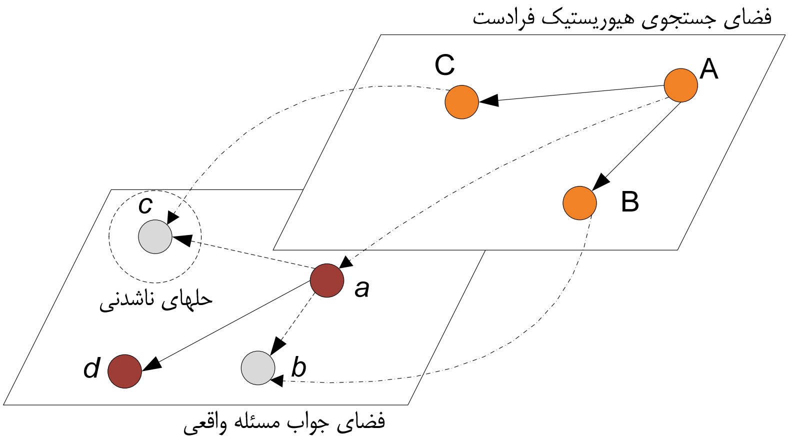
در روش‌های زمان‌بندی براساس جست‌وجوی محلی، حرکت به سوی همسایهٔ بعدی با تغییر بازهٔ زمانیِ تعدادی از رویدادها در جواب واقعی انجام می‌شود. در هیوریستیک سطح بالا در GbHH با حرکت به سوی همسایگی یک توالی از هیوریستیک‌ها، جواب‌های ساخته‌شده این توانمندی را دارند که بسیار متفاوت از یکدیگر باشند. این متفاوت بودن جواب‌ها، تضمین‌کنندهٔ پوشش بخش‌های مختلف از فضای جواب مسئله است. جست‌وجو به‌کمک هیوریستیک سطح بالا را می‌توان یک جهش در فضای جواب درنظر گرفت. با شمار محدودی حرکت با جست‌وجوی هیوریستیک سطح بالا، جواب‌های به‌دست‌آمده در فضای جواب پخش می‌شوند.

هیوریستیک سطح بالا

در به‌کارگیری روش GbHH دو هدف دنبال می‌شود. نخست، هیوریستیک سطح بالا در چارچوب GbHH با اطلاعات مربوط به مسئلهٔ اصلی کار نخواهد کرد، بلکه روی مقدارهای تابع هدف جست‌وجو می‌کند که از هیوریستیک‌های سطح پایین به‌دست می‌آید. همهٔ اطلاعات مورد نیاز دربارهٔ مسئله، به‌کمک هیوریستیک‌های سطح پایین بررسی می‌شود. این امر باعث ایجاد

کلیت در GbHH می‌شود و آن را برای استفاده در مسئله‌های دیگر آسان می‌کند. دوم، استفاده از هیوریستیک‌های ساده در سطح پایین، نیاز به بیان مسئله به شکلی خاص را از میان برمی‌دارد. این موضوع نیز به جامعیت روش GbHH کمک می‌کند.

در شکل ۲، در لایهٔ بالا سه هیوریستیک A، B و C جای دارند که هر کدام در فضای جواب مسئله به یک حل a، b یا c نگاشت یافته است. در لایهٔ پایین، حل‌های a و b که با هیوریستیک‌های A و B در لایهٔ بالا مطابقت دارند، در همسایگی هم در فضای جست‌وجوی حل‌ها هستند. هیوریستیک فرادست با تغییر در هیوریستیک‌های حل در لایهٔ بالا، می‌تواند در فضای حل مسئلهٔ واقعی از یک حل به حل دیگر پرش کند. با به‌کارگیری این ترفند، افزون‌بر افزایش سرعت جست‌وجو، منطقهٔ بیشتری از فضای حل مسئله نیز جست‌وجو می‌شود.



شکل ۲. رابطه میان فضای جست‌وجوی هیوریستیک فرادست و فضای حل مسئله

در این مقاله یک روش سادهٔ تندترین شیب به‌کار برده می‌شود که در آن، بهترین توالی هیوریستیک در میان همسایه‌ها اگر بهتر یا همانند جواب فعلی باشد، انتخاب می‌شود. جست‌وجو هنگامی متوقف می‌شود که هیچ همسایهٔ بهتر یا همانند وجود نداشته باشد، یا اینکه شمار تکرار و ارزیابی به تعداد از قبل تعیین‌شده رسیده باشد.

همچنین ساختارهای همسایگی ساده‌ای با تغییر تصادفی ۲، ۳، ۴ یا ۵ هیوریستیک در فهرست با هیوریستیک‌های رنگ‌آمیزی دیگر تعریف می‌شود. این ساختارهای سادهٔ همسایگی، کلی و برای اجرا آسان هستند.

یافته‌های پژوهش

در این مقاله برای جست‌وجوی محلی در GbHH از ترکیب‌سازی (اجرای) جست‌وجوی محلی استفاده می‌شود که یک جست‌وجوی سادهٔ آزمندانه است. در هر مرحله از هیوریستیک سطح بالا، یک جواب کامل به‌کمک فهرست هیوریستیک ایجاد خواهد شد. سپس برای بهبود کیفیت جواب به‌سمت بهینگی محلی، یک جست‌وجوی آزمندانه روی جواب کامل، درون فضای جواب اجرا خواهد شد؛ بدین صورت که یکی از رویدادهایی که تولید امتیاز منفی می‌کند انتخاب‌شده، سپس در بازه‌های زمانی ممکن جای‌گذاری و تابع هدف بررسی می‌شود تا در بهترین بازه‌ای قرار گیرد که تابع هدف را بهبود می‌دهد. این شیوه منجر به بهبود محلی روی جواب کامل می‌شود.

بدین ترتیب، نه‌تنها توالی‌های مختلف در فضای جست‌وجوی هیوریستیک‌ها، برای رسیدن به بهترین جواب های ممکن در فضای جواب جست‌وجو می‌شوند، بلکه جواب‌های به‌دست‌آمده نیز به‌کمک جست‌وجوی محلی در فضای جواب، بهبود می‌یابند. جست‌وجوی دوم همانند بیشتر الگوریتم‌های جست‌وجوی محلی ارائه‌شده در پژوهش‌های پیشین است که روی فضای جست‌وجوی جواب کار می‌کنند.

رویکرد پیشنهادی GbHH روی داده‌های یک مسئلهٔ واقعی برای زمان‌بندی کلاس‌های دانشگاه خلیج فارس اجرا شده است. اطلاعاتی از مسئله که برای تنظیم یک جدول زمان‌بندی لازم است، در زیر گردآوری شده‌اند:

- دانشگاه فهرستی از درس‌های تفکیک‌شده در هشت نیمسال تحصیلی را به دانشجو پیشنهاد می‌دهد. دانشجو باید بر اساس برنامهٔ پیشنهادی، انتخاب واحد را انجام دهد. برای اینکه دانشجو بتواند بر اساس این پیشنهاد عمل کند، درس‌های یک نیمسال نباید همپوشی داشته باشند. بدین منظور، نخست رشته‌ها و سپس گروه‌های درسی هر رشته (نیمسال یک، نیمسال دو، ...) معرفی می‌شوند. زمان برگزاری درس‌های یک گروه نباید همپوشی داشته باشند. این در واقع یک محدودیت سخت است؛
- با فرض اینکه درس‌ها به استادان تخصیص داده شده‌اند و هیچ درسی بدون استاد نیست، دو محدودیت سخت وجود دارد. نخست، هر درس در زمان پیشنهادی استاد برگزار شود. دوم، زمان برگزاری درس‌های یک استاد باید همپوشی نداشته باشند؛
- هر درس ارائه‌شده می‌تواند یک، دو یا سه جلسه‌ای باشد؛ بدین معنا که در هفته به تعداد مشخصی برگزار می‌شود. در یک روز حداکثر یک جلسه از یک درس می‌تواند برگزار شود. این نیز یک محدودیت سخت است؛

- تعداد اتاق‌ها و گنجایش هر اتاق، داده‌های مورد نیاز از اتاق‌هایی است که مکان برگزاری کلاس خواهند بود. در یک بازهٔ زمانی نباید بیش از یک کلاس به هر اتاق تخصیص داده شود و ظرفیت اتاق باید بزرگتر یا مساوی شمار دانشجویان درس تخصیص داده‌شده به آن اتاق باشد. این نیز یک محدودیت سخت است.

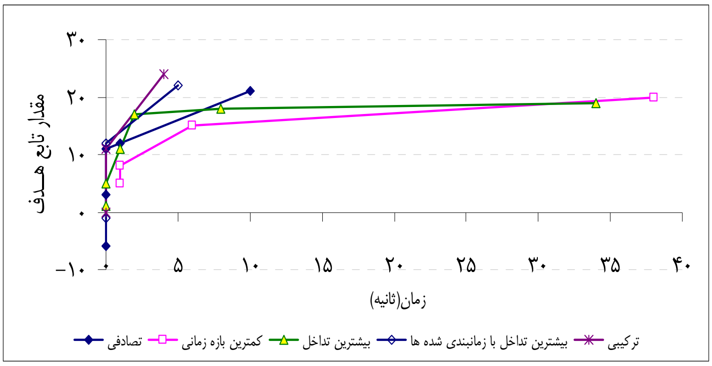
همان‌گونه که پیش از این گفته شد، باید در هر حل این محدودیت‌های سخت برقرار باشند. یک حل هنگامی یک جواب شدنی و قابل قبول است که محدودیت‌های سخت را برآورده کند. افزون‌بر محدودیت‌های سخت، دستهٔ دیگری از محدودیت‌ها در نظر گرفته شده‌اند که باید تلاش شود تا حد امکان برآورده شوند. میزان برآورده‌سازی آنها بیان‌کنندهٔ ارزش تابع هدف سطح بالا (کیفیت جواب) است. محدودیت‌های نرم عبارتند از:

- میان جلسه‌های هفتگی یک درس، دست‌کم یک روز فاصله باشد؛
- هیچ جلسه‌ای در آخرین بازهٔ زمانی یک روز تشکیل نشود؛
- در یک روز، دست‌کم دو و حداکثر سه کلاس برای یک گروه تشکیل شود؛
- جلسه‌های برگزارشده برای یک گروه در یک روز، پیاپی و بدون فاصله باشد.

برای هر کدام از این محدودیت‌های نرم یک وزن درنظر گرفته شده است که نشان‌دهندهٔ اهمیت محدودیت مربوطه است. برای حل این مسئلهٔ واقعی، بستهٔ نرم‌افزاری GbHHCT با زبان برنامه‌نویسی سی شارپ (C#) و به‌کمک یک پایگاه داده پیاده‌سازی‌شده در MS Access تولید شد. رشته‌ها، استادان و زمان پیشنهادی آنان، اطلاعات مربوط به گروه‌های درسی هر رشته، درس‌های ارائه‌شده و اطلاعات مربوط به آنها و نیز، اتاق‌های موجود و گنجایش آنها، داده‌های ورودی بستهٔ نرم‌افزاری است.

پس از ورود داده‌ها، در بخش روش‌های انتخاب می‌توان هر یک از هیوریستیک‌ها یا هیوریستیک فرادست را انتخاب کرد. با استفاده از تنظیم پارامترهای اجرا، برنامه برای مدت‌زمان یا شمار تکرار از پیش تعریف‌شده، اجرا می‌شود. نتایج به‌دست‌آمده از نظر تعداد تکرارهایی که به جواب رسیده، مقدار تابع هدف و مدت‌زمان رسیدن به بهترین جواب با هم مقایسه می‌شود.

شکل ۳ اجرای پنج روش مختلف در مدت‌زمان یک دقیقه را با هم مقایسه می‌کند. همان‌گونه که دیده می‌شود، در مقایسه با دیگر هیوریستیک‌ها، هیوریستیک فرادست در مدت‌زمان کمتر جواب بهتری را تولید کرده است. پس از اجرای برنامه، می‌توان برون‌داد تولیدشده را در چارچوب برنامهٔ هفتگی و به‌تفکیک هر گروه مشاهده کرد.



شکل ۳. نمودار مقایسهٔ نتایج اجرای روش‌های ابتکاری در بازهٔ یک دقیقه

برنامه GbHHCT، پس از پردازش، برای هر یک از گروه‌های آموزشی و به‌ازای ورودی‌های هر سال آموزشی، برنامهٔ هفتگی را تولید می‌کند. نمونه‌ای از بـرون‌داد برنامـه در شـکل ۴ نشـان داده شده است.

روز / ساعت	۸ الي ۹,۵	۱۰ الي ۱۱,۵	۱ الي ۲,۵	۳ الي ۴,۵	۵ الي ۶,۵
شنبه	رياضيات كاربرد || كلاس:۲	اقتصاد خرد || كلاس:۳	حسابداري صنعتي۱ || كلاس:۲	----	----
يك شنبه	روش تحقيق || كلاس:۱	آمار ۲ || كلاس:۲	----	----	----
دو شنبه	حسابداري صنعتي۱ || كلاس:۲	روش تحقيق || كلاس:۲	انديشه اسلامي || كلاس:۱	رياضيات كاربرد || كلاس:۲	----
سه شنبه	اقتصاد خرد || كلاس:۲	آمار ۲ || كلاس:۲	زبان۱ || كلاس:۲	----	----
چهار شنبه	----	----	----	----	----
پنجشنبه	----	----	----	----	----

شکل ٤. یکی از برون‌دادهای برنامه

نتیجه‌گیری و پیشنهادها

حل یک مسئلهٔ زمان‌بندی کلاس‌های دانشگاه، بسیار سخت و زمـان‌بـر و از گونـه مسـئله‌هـای NP-hard است. در این نوشتار، برای زمان‌بنـدی کـلاس‌هـای دانشـگاه، رویکـرد هیوریسـتیک فرادست برمبنای رنگ‌آمیزی گراف پیشنهاد شد. سودمندی این رویکرد این است که مدل‌سـاز را از بیان مسئله به شکل خاص بی‌نیاز می‌کند. همچنین این رویکرد، مسئله را در مدت زمان بسیار

کوتاه حل می‌کند و از هیچ‌یک از محدودیت‌های سخت چشم‌پوشی نمی‌کند. حل یک مسئلهٔ واقعی نشان داد که جواب به‌دست‌آمده در مقایسه با زمان‌بندی انجام‌شده به‌صورت دستی بسیار بهتر است.

از آنجاکه مسئلهٔ زمان‌بندی در دانشگاه با توجه به نیازهای خاص مسئله می‌تواند بسیار متفاوت باشد، از این رو پژوهشگران می‌توانند با افزودن محدودیت‌های دیگر، سازگاری رویکرد پیشنهادی در این مقاله را با شرایط مسئلهٔ واقعی بیشتر کنند. این محدودیت‌ها می‌تواند در نظر گرفتن زمان پیشنهادی استادان، درنظرگرفتن محدودیت لزوم اتاق خاص برای برخی درس‌ها، یا پیاپی بودن برنامهٔ کلاس‌های هر استاد باشد. این‌گونه محدودیت‌ها را می‌توان محدودیت‌های نرم تعریف کرد و سپس به مدل مسئله افزود.

منابع

علیرضایی، م.؛ خلیلی، م. و منصورزاده، م. (۱۳۸۵). برنامه‌ریزی درسی در دانشگاه به کمک مدل‌سازی دو مرحله‌ای برنامه‌ریزی ریاضی. *دو ماهنامهٔ دانشور رفتار*، ۱۳(۱۷): ۹۶- ۸۷.

Abramson, D. (1991). Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms. *Management Science,* 37(1): 98-113.

Abramson, D., Dang, H. & Krisnamoorthy, M. (1999). Simulated Annealing Cooling Schedules for the School Timetabling Problem. *Asia-Pacific Journal of Operational Research,* 16: 1-22.

Agustín-Blas, L., Salcedo-Sanz, S., Ortiz-García, E., Portilla-Figueras, A. & Pérez-Bellido, Á. (2009). A Hybrid Grouping Genetic Algorithm for Assigning Students to Preferred Laboratory Groups. *Expert Systems with Applications,* 36(3): 7234-7241.

Asmuni, H., Burke, E. & Garibaldi, J. (2005). Fuzzy Multiple Ordering Criteria for Examination Timetabling. In E. Burke & M. Trick, *Lecture Notes in Comouter Science,* 3616: 334-353.

Bilgin, B., Özcan, E. & Korkmaz, E. (2007). An Experimental Study on Hyper-Heuristics and Exam Scheduling. In E. Burke & H. Rudova, *Lecture Motes in Computer Science,* 3867: 394-412.

Brucker, P. & Knust, S. (2006). *Complex Scheduling.* Springer. Berlin

Burke, E. & Petrovic, S. (2002). Recent Research Directions in Automated Timetabling. *European Journal of Operational Research,* 140(2): 266-280.

Burke, E., Dror, M., Petrovic, S. & Qu, R. (2005). Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems. In B. Golden, S. Raghavan & E. Wasil, *The Next Wave in Computing, Optimization and Decision Techniques,* 29: 79-91.

Burke, E., Elliman, D. & Weare, R. (1994). A Genetic Algorithm Based University Timetabling System. *Proceedings of the 2nd East-West International Conference on Computer Technologies in Education*, Crimea, Ukraine. 1: 19-23.

Burke, E., Hart, E., Kendall, G., Newall, J., Ross, P. & Schulenburg, S. (2003). Hyperheuristics: an Emerging Direction in Modern Search Technology. In F. Glover & G. Kochenberger, *Handbook of Meta-Heuristics,* 457-474.

Burke, E., Kendall, G. & Soubeiga, E. (2003). A Tabu Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics,* 9(6): 451-470.

Burke, E., Kingston, J. & de Werra, D. (2004). Applications to Timetabling. In J. Gross & J. Yellen, *Handbook of Graph Theory.* Chapman Hall/CRC Press.

Burke, E., McCollum, B. & Meisels, A. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research,* 176(1): 177-192.

Burke, E., Newall, J. & Weare, R. (1996). A Memetic Algorithm for University Exam Timetabling. In E. Burke & P. Ross (Ed.), *Practice and theory of automated timetabling: Selected papers from the first international conference*, *1153,* Edinburgh, UK, August 29 - September 1.

Burke, E., Petrovic, S. & Qu, R. (2006). Case Based Heuristic Selection for Examination Timetabling. *Journal of Scheduling,* 9(2): 99-113.

Carrasco, M. & Pato, M. (2004). A Comparison of Discrete and Continuous Neural Network Approaches to Solve the Class/teacher Timetabling Problem. *European Journal of Operational Research,* 153 (1): 65-79.

Carter, M. & Laporte, G. (1998). Recent developments in practical course timetabling. In E. Burke & M. Carter (Eds.), *Practice and Theory of Automated Timetabling II, Lecture Notes in Computer Science,* 1408: 3-19.

Eckersley, A. (2007). *Novel Knowledge based and Heuristic Approaches to University Timetabling.* School of Computer Science and Information Technology, University of Nottingham.

Gaw, A., Rattadilok, P. & Kwan, R. (2005). Distributed Choice Function Hyperheuristics for Timetabling and Scheduling. In E. Burke & M. Trick (Eds.), *Lecture Notes in Computer Science,* 3616: 51-67.

Hertz, A. (1991). Tabu Search for Large Scale Timetabling Problems. *European Journal of Operational Research,* 54: 39-47.

Lewis, R., Paechter, B. & Rossi-Doria, O. (2007). Metaheuristics for University Course Timetabling. In *Evolutionary Scheduling,* 49: 237–272.

MirHassani, S. (2006). A Computational Approach to Enhancing Course Timetabling With Integer Programming. *Applied Mathematics and Computation,* 175 (1): 814-822.

MirHassani, S. (2006). Improving Paper Spread in Examination Timetables using Integer Programming. *Applied Mathematics and Computation,* 179 (2): 702-706.

Pongcharoen, P., Promtet, W., Yenradee, P. & Hicks, C. (2008). Stochastic Optimisation Timetabling Tool for University Course Scheduling. *International Journal of Production Economics,* 112(2): 903-918.

Qu, R. & Burke, E. (2009). Hybridisations within a Graph Based Hyper-heuristic Framework for University Timetabling Problems. *Journal of the Operational Research Society,* 60: 1273-1285.

Ross, P., Corne, D. & Fang, H. (1994). *Improving Evolutionary Timetabling with Delta Evaluation and Directed Mutation.* PPSN III, Springer Verlag.

Ross, P., Marin-Blazquez, J. & Hart, E. (2004). Hyper-heuristics Applied to Class and Exam Timetabling Problems. *Proceedings of the congress on evolutionary computation (CEC2004)*, 2: 1691-1698. Edinburgh, UK.

Schaerf, A. (1999, April). A Survey of Automated Timetabling. *Artificial Intelligence Review,* 13(2): 84-127.

Schaerf, A. & Schaerf, M. (1995). Local Search Techniques for High School Timetabling. *Proceedings of the 1st Intl. Conference on the Practice and Theory of Automated Timetabling*, 313-323, Edinburgh, UK, August 29 - September 1.

Terashima-Marin, H., Ross, P. & Valenzuela-Rendon, M. (1999). Evolution of Constraint Satisfaction Strategies in Examination Timetabling.

Proceedings of the Genetic and Evolutionary Computation Conference, 635-642. 13-17 July, Orlando, Florida, USA.

Van den Broek, J., Hurkens, C. & Woeging, G. (2009). Timetabling Problems at the TU Eindhoven. *European Journal of Operational Research,* 196(3): 877–885.