وزارت علوم، تحقیقات و فناوری

دانشکده ریاضی

سمینار درس بهینه سازی ترکیبیاتی

زمانبندی کلاسهای دانشگاه

استاد

دکتر قنبری

پژوهشگر

مهدی رضاپور

زمستان ۱۳۹۰

فهرست مطالب

۱ مقدمه

زمانبندی به بخش مورد علاقه روز افزون در جامعه برای تحقیق و عمل در دهه های اخیر تبدیل شده است. موارد نمونه در این زمینه عبارتند از: زمانبندی آموزشی (دانشگاه،مدرسه) [۳] ، زمانبندی ورزش [۴] زمانبندی کارمندان [۵] زمانبندی حمل و نقل [۶] و غیره.

زمانبندی آموزشی دانشگاه به دو دسته طبقه بندی شده است زمانبندی امتحان و زمانبندی کلاسهای درس.در این سمینار، ما زمانبندی کلاسهای درس دانشگاه را برسی میکنیم.

در حالت کلی، مسائل ترکیبیاتی و بهینه سازی را میتوان به صورت زیر دسته بندی کرد:

مسائل ارضای محدودیت :

فقط یک راه حل از بین کلیهٔ راه حل ها که در تمامی محدودیت ها صدق کند کافی است بدون هیچ گونه اولویتی میان راه حل ها .

مسائل بهینه سازی :

هدف یافتنِ بهترین راه حلی است که در تمامی محدودیتها صدق کند و یک تابع هدف را کمینه یا بیشینه کند .

مسائل فوق العاده محدود :

حلی که در تمامی محدودیتها صدق کند یا موجود نیست یا با هزینهٔ زیاد به دست می آید (اغلب به دلیل زیادی تعداد محدودیتها). در این صورت محدودیتها را به دو دسته تقسیم میکنند:

محدودیتهای سخت: حتماً باید ارضا شوند:

- زمانی که یک استاد در دانشگاه حضور ندارد برای اوکلاسی نگذاریم .
- دو درس یک ورودی نباید در یک ساعت باشد.

محدودیتهای نرم : میتوانند ارضا نشوند اما بهتر است ارضا شوند.

- مثلاً یک استاد ساعتهای خاصی را بیشتر ترجیح می دهد.
- فاصله درسهای یک استاد (دانشجو) در یک روز زیاد نباشد.

در اینگونه مسائل، حل نهایی باید تمام محدودیتهای سخت و حداکثر تعدادِ ممکن از محدودیتهای نرم را ارضا کند.

زمان بندی کلاسهای درس از مسائل فوقالعاده محدود است که تا به امروز حل کاملی برای آن ارائه نشده است در این مساله سعی بر این است که مجموعه ای از منابع معین ، متشکل از کلاسها ، اساتید و دروس تحت شرایط خاص به مجموعه ای از ساعتهای درسی اختصاص یابد .

۲ روشهای حل

مسأله timetabling از حدود ۵۰ سال پیش توسط Gotlieb (1962) مطرح شد. از آن زمان تا به حال روشهای مختلفی برای حل این مساله بکار گرفته شده است که در اینجا ما به تعدادی ازاین روشها [۱] اشاره میکنیم :

۱. **روشهای براساس برنامه ریزی ریاضی :** مانند ILP اما این روشها به دلیل زمان زیاد محاسبات جذابیت کمتری دارند.

۲. **روشهای مبتنی بر رنگ آمیزی گراف :**

مسأله زمانبندی می تواند به وسیله گرافها نشان داده شود که رئوس نشان دهنده دروس است در حالی که یالها نشان دهنده تداخل بین دروس است.

در این قسمت روشهای مختلفی را برای ساخت یک جدول زمانی بدو ن تداخل بکار می گیرند. این روشها دروس را بر اساس میزان سختی تخصیص زمانبندی آنها، برخی از این روش ها به صورت زیر است:

- **اولین درجه بزرگ :** درسهایی با بیشترین تعداد تداخل زودتر برنامه ریزی می شوند.
- **بزرگترین درجه وزنی :** تفاوتی که با روش قبلی دارد این است که به هر تداخل وزنی نسبت داده می شود که بیانگر تعداد دانشجویان آن تداخل است.
- **بزرگترین درجه وزنی :** درجه اشباع : در هر مرحله از ساخت زمانبندی، درسی را که کمترین تعداد دوره های معتبر را در زمانبندی ساخته شده فعلی در اختیار دارد انتخاب می شود.
- **درجه رنگ :** در این جستجو وقایعی که بیشترین تعداد تداخل را با رویداد های برنامه ریزی دارند ، انتخاب می شوند.

سپس می توان راهکارهای متعددی را برای انتخاب بازه زمانی برای هر رویداد بکار برد.

(این روشها برای نمونه های کوچک خیلی جالب و مفید هستند ولی برای نمونه های بزرگ کاربرد ندارند).

[۱] ، [۲]

۳. **روشهای دسته کردن (خوشه ای) :**

در این روش درسها را طوری گروه بندی می کنند که قیود سخت برآورده شوند پس گروهها را به پریودهای زمانی تخصیص می دهند تا قیود نرم را برآورده کنند اما این روش به جوابهایی با کیفیت پائین منتهی می شود.

۴. **روشهای مبتنی بر قیود :** در این روش ، دروس مدل می شوند به وسیله مجموعه ای از متغیرها با مقادیری که باید به آنها تخصیص داده شوندمشروط بر تعدادی قیود. وقتی که تعمیم مقادیر تخصیص داده شده منتهی به جوابی نشدنی شود، یک پروسه بازگشت به عقب قادر است دوباره مقادیری تخصیص دهد تا اینکه جوابی که همه قیود را برآورده کند، پیدا شود.

۵. **روشهای متاهیورستیک :**

- الگوریتم ژنتیک (Genetic algorithms)
- جستجوی ممنوع (Tabu search)
- انجماد تدریجی (Simulated annealing)
- الگوریتم مورچگان (Max -Min Ant System)

که از میان آ نها ما در ادامه دو روش اول را برسی می کنیم .

۶. **روشهای تعقل بر پایه مورد :** روشهای نسبتاً جدیدی هستند که بر اساس استفاده از تجربیات گذشته استوار هستند و به دو صورت برای حل مسأله زمانبندی استفاده می شوند:

- به عنوان یک تکنیک استفاده مجدد جواب.

در این حالت سیستم یک مورد که بیشترین شباهت را به مسأله داشته می یابد.

- متدلوژی : در این مورد سیستم روشی را پیشنهاد می کنند که برای حل مسأله مناسب تر است.

۷. **روشهای چند معیاره :** در این روشها هر معیار نقض قید متناظر را اندازه گیری می کنند. معیارها در موقعیت های مختلف و برای مراکز مختلف، سطوح متفاوتی دارند.

۳ مدل ILP

مدلهای ILP متنوعی برای مساله $CourseTimeTabling$ ارائه شده است مدلی که ما ارائه می کنیم مربوط می شود به $(deWerra۱۹۸۵)$:

فرض کنید n درس $C = \{c_۱, c_۲, ..., c_n\}$ داریم و برای هر i درس c_i شامل l_i سخنرانی است و $CR = \{Cr_۱, Cr_۲, ..., Cr_s\}$ یک مجموعه از گروه های آموزشی است که در آن هر Cr_k یک گروه از درس هایی است که دانشجوی مشترک دارند .و فرض کنید که p تعداد دوره ها و R_k تعداد دوره های در دسترس در دوره k باشد در این صورت داریم :

$$\max \sum_{i=۱}^{n} \sum_{k=۱}^{p} d_{i,k} y_{i,k}$$

$s.t.$

$$\sum_{k=۱}^{p} y_{i,k} = l_i \qquad (i = ۱, ..., n)$$

$$\sum_{i=۱}^{n} y_{i,k} \leqslant R_k \qquad (k = ۱, ..., p)$$

$$\sum_{i \in Cr_l} y_{i,k} \leqslant ۱ \qquad (l = ۱, ..., s, k = ۱, ..., p)$$

$$y_{i,k} = ۰ \; or \; ۱ \qquad (i = ۱, ..., n, k = ۱, ..., p)$$

where

$$y_{i,k} = \begin{cases} ۱ & if\ a\ lecture\ of\ course\ C_i\ is\ scheduled\ at\ period\ k \\ ۰ & otherwise. \end{cases}$$

که درآن $d_{i,k}$ ارزش برگزار شدن درس i ام در دوره k ام می باشد.

۴ پیچیدگی

برای اثبات NP-Hard مساله $CourseTimeTabling$ ثابت می کنیم که این مساله به مساله رنگ آمیزی راسی گرافها کاهش پذیر است . متناظر با هر سخنرانی(جلسه) از هر درس یک راس در گراف G نسبت می دهیم . و بین هر دو راس یال وجود دارد اگر وفقط اگر دو درس متناظر با آن دو سخنرانی در یک گروه درسی باشند. در این صورت مساله k –رنگ پذیری راسی گراف G متناظر با این مساله است که آیا می توانیم n درس ارائه شده را در k دوره برنامه ریزی کنیم یا نه؟ .

۵ جستجوی ممنوع

طرح کلی الگوریتم پیشنهادی ATS [۲] زیر متشکل از سه مرحله است: مقداردهی اولیه، تشدید و تنوع.

در مرحله مقداردهی اولیه یک جدول زمان بندی شدنی با استفاده از یک روش سریع حریصانه ساخته می شود سپس به منظور کاهش تعداد نقض محدودیت نرم در حالی که محدودیت های سخت شدنی باقی بمانند مراحل تشدید و تنوع به طور سازگار با هم ترکیب می شوند.

۱.۵ شرح مساله

در مساله CTT(Course Time Tabling) هدف زمانبندی سخنرانی های (جلسات) یک مجموعه از دروس در داخل یک جدول زمانبندی هفتگی است بطوری که هر جلسه از یک درس باید در یک دوره و یک اتاق مجزا بر طبق یک مجموعه از محدودیت های داده شده قرار گیرد . یک جدول زمانبندی را شدنی گوییم هرگاه همه سخنرانی ها (جلسات) در یک دوره و یک اتاق مجزا طوری قرار گیرندکه محدودیت های سخت $H_۱ - H_۴$ ارضا شوند . بعلاه هر جدول زمانی شدنی یک هزینه جریمه برای نقض محدودیتهای نرم $S_۱ - S_۴$ را تحمیل می کند و در نهایت هدف مساله CTT مینیمم کردن تعداد محدودیتهای نقض شده در یک جدول زمانبندی شدنی می باشد.

$H_۱$: دربرنامه هفتگی تعداد جلسات هر در س رعایت شده باشد.

$H_۲$: هیچ دو سخنرانی نباید در یک دوره به یک کلاس یکسان نسبت داده شوند.

$H_۳$: در یک دوره برای یک استاد و یا دانشجو در دو کلاس برنامه ریزی نشود .

$H_۴$: اگر در یک دوره ای یک استادی در دسترس نیست برای او برنامه ریزی نشود .

$S_۱$: برای هر سخنرانی تعداد دانشجویان حاضر در آن دوره نباید بیشتر از ظرفیت کلاس باشد .

$S_۲$: تمام سخنرانی های یک درس باید باید در یک کلاس یکسان برنامه ریزی شوند .

$S_۳$: جلسات یک درس باید در حداقل روز کاری داده شده ، پخش شوند .

$S_۴$: اگر دو درس متعلق به یک گروه درسی هستند و در یک روز ارائه می شوند پشت سر هم باشند .

[۲] [۹]

۲.۵ فرمولبندی مساله

مساله CTT از یک مجموعه n تایی از درسهای $C = \{c_۱, c_۲, ..., c_n\}$ تشکیل شده است که باید در یک مجموعه p تایی از دوره های $T = \{t_۱, t_۲, ..., t_p\}$ و یک مجموعه m تایی از اتاقهای $R = \{r_۱, r_۲, ..., r_m\}$ برنامه ریزی شود . هر درس c_i شامل l_i سخنرانی (جلسه) می باشد که باید برنامه ریزی شود . برای سادگی و تا زمانی که مشکلی به وجود نیاید ما هیچ تفاوتی بین اندیس درس و سخنرانی آن قائل نمی شویم .

یک دوره یک جفتی است متشکل از یک روز و یک $timeslot$ (یکی از بازه های زمانی تدریس در طول روز) ، و $CR = \{Cr_۱, Cr_۲, ..., Cr_s\}$ یک مجموعه از گروه های آموزشی است که در آن هر Cr_k یک گروه از درس هایی است که دانشجوی مشترک دارند .

یک جواب کاندید را با یک ماتریس $X_{p\times m}$ نمایش می دهیم که در آن x_{ij} متناظر با درسی است که در دوره t_i به اتاق r_j نسبت داده شده است . این شکل نمایش ما برقراری محدودیت سخت $H_۲$ را تضمین میکند .

- $\forall c_k \in C : H_۱$

$$\sum_{i=۱,...,n, j=۱,...,m} \chi_{\{x_{ij}=c_k\}} .$$

که در آن χ تابع مشخصه است .

- $H_۲$: به دلیل نوع نمایش جواب همیشه برقرار است .

- $\forall x_{ij}, x_{ik} \in X, x_{ij} = c_u, x_{ik} = c_v : H_۳$

$$con_{uv} = ۰.$$

- $\forall x_{ij} = c_k \in X : H_۴$

$$uav_{k,i} = ۰.$$

- $\forall x_{ij} = c_k \in X : S_۱$

$$f_۱(x_{i,j}) = \begin{cases} std_k - cap_j, & if \quad std_k > cap_j \\ ۰ & \text{otherwise.} \end{cases},$$

- $\forall c_i \in C : S_۲$

$$f_۲(c_i) = nr_i(X) - ۱.$$

با توجه به فرمولبندی فوق ما می توانیم هزینه کل نقض محدودیتهای نرم را برای یک جواب کاندید X بر طبق فرمول زیر محاسبه کنیم.

$$f(X) = \sum_{x_{i,j} \in X} \alpha_۱ . f_۱(x_{i,j}) + \sum_{c_i \in C} \alpha_۲ . f_۲(c_i) + \sum_{c_i \in C} \alpha_۳ . f_۳(c_i) + \sum_{x_{i,j} \in X} \alpha_۴ . f_۴(x_{i,j}) \qquad (۱)$$

که در آن $\alpha_۴, \alpha_۳, \alpha_۲, \alpha_۱$ جریمه های متناظر با هر یک از محدودیتهای نرم می باشند. پس هدف یافتن یک جواب شدنی X^* می باشد به طوری که برای همه X شدنی در فضای جستجو داشته باشیم : $f(X^*) \leq f(X)$.

- $\forall c_i \in C : S_۳$

$$f_۳(c_i) = \begin{cases} md_i - nd_i(X), & if \quad nd_i(X) < md_i, \\ \\ ۰ & \text{otherwise.} \end{cases}$$

- $: S_۴$

$$\forall x_{ij} = c_k \in X$$

$$f_۴(x_{i,j}) = \sum_{Cr_q \in CR} \chi_{\{c_k \in Cr_q\}} . iso_{q,i}(X),$$

$$where$$

$$iso_{q,i}(X) = \begin{cases} ۱, & \text{if (i mod h = ۱} \bigvee app_{q,i-۱}(X) = ۰) \\ & \bigwedge (i \ mod \ h = ۰ \bigvee app_{q,i+۱}(X) = ۰) \\ \\ ۰, & otherwise. \end{cases}$$

۳.۵ مقداردهی اولیه

مقدار دهی اولیه را با یک جدول خالی و الگوریتم حریصانه آغاز می کنیم. و برای تخصیص درس ها یک سخنرانی مناسب را در یک زمان خالی از جدول قرار می دهیم. در هرگام دو عملیات مجزا اجرا می شود. یکی انتخاب یک سخنرانی علامت نخورده از یک درس و دیگری مشخص کردن دوره واتاق برای این سخنرانی. ابتکاری که برای انتخاب سخنرانی انجام می دهیم این است که درس هایی که استاد آن ها را در دوره های باقی مانده جدول کمتر در دسترس است. و تعداد جلسات (سخنرانی های) علامت نخورده آن ها بیشتر است را زودتر انتخاب می کنیم.

۴.۵ الگوریتم جستجوی تابو (ATS)

در این بخش ما متمرکز می کنیم روی اصول جستجوی الگوریتم ATS خودمان و TS [۱۰].

روال TS ما از دو همسایگی استفاده می کند. (که با $N۱$ و $N۲$ نشان داده شده اند) در یکی از آن ها به روش رینگ گرفته می شود [۱۱].

به طور دقیق تر، روال TSرا با یک همسایگی شروع می کنیم زمانی که جستجو در بهترین بهینه محلی خاتمه یافت، دوباره TS را از این بهینه محلی با استفاده از همسایگی دیگر ادامه می دهیم. این روال را تا زمانی که نتوانیم بهبود ایجاد کنیم ادامه می دهیم در این حالت است که یک فازTS تمام می شود. در مورد کار ما روال TS شروع می شود با $N۱$:

$$N۱ \rightarrow N۲ \rightarrow N۱ \rightarrow N۲$$

۵.۵ جستجوي فضا و تابع ارزياب

به محض اينكه به يك جدول زماني ممكن كه در همه محدوديت هاي سخت صدق كند برسيم، فاز تشديد ما (الگوريتم TS) تابع هزينه محدوديت هاي نرم را بدون نقض محدوديت هاي سخت، تا حد ممكن بهينه مي كند. بنابراين فضاي جستجوي الگوريتم TS مان به جدول زماني هاي ممكن (جواب هاي شدني) محدود شده است. تابع ارزياب همانطور

که در فرمول (۱) مشخص شده است فقط براي نقض محدوديت هاي نرم است.

۶.۵ ساختارهاي همسايگي

خيلي ها معتقدند که يکي از مهم ترين ويژگي هاي الگوريتم $LS(LocalSearch)$ تعريف همسايگي آن می باشد در يك روال LS بکارگيري يك حرکت mv که يکي از کانديدها براي جواب X است منجر به يك جواب جديدي مي شود که آن را با $X \oplus mv$ نشان مي دهند. فرض کنيد $M(X)$ مجموعه تمام حرکت هاي شدني باشد که مي توانيم در X استفاده کنيم به طوري که شدني بمانيم، آن گاه همسايگی N از X به صورت زير تعريف مي شود.

$$\mathbf{N(X)} = \{X \oplus mv | mv \in M(X)\}$$

ما براي مساله CTT از دو نوع حرکت مجزا که با $simpleswap$ و $kempeswap$ نمايش مي دهيم، استفاده مي کنيم.

همسايگی ساده $N۱$: $N۱$ تشکيل شده است از همه حرکت هاي شدني $simpleswap$.

يك حرکت $simpleswap$ متشکل از تبادل ميزباني دوره ها و اتاق هاي تخصيص داده شده به دو جلسه (سخنراني) از درس هاي مختلف است. با بکارگيري حرکت $simpleswap$ در دو درس مختلف $x_{i,j}$ و $x_{i',j'}$ در جواب X باعث مي شود که مقدار $x_{i,j}$ به $x_{i',j'}$ و برعکس مقدار $x_{i',j'}$ به $x_{i,j}$ تخصيص يابد. توجه کنيد که حرکت يك جلسه از يك درس به يك مکان خالي يك حالت خاص از حرکت $simpleswop$ است.

بنابراين اندازه همسايگي $N۱$ محدود شده است به وسيله $o(l.p.m)$ که در آن $l = \sum_{i=۰}^{n-۱} l_i$

(l_i تعداد جلسات درس c_i است ، P تعداد کل دوره ها، m تعداد کل کلاسها) زيرا در کل l جلسه داريم و تعداد جابجايي جلسات (شامل مکان هاي خالي) به وسيله $o(p.m)$ محدود مي شود.

همسايگي پيشرفته $N۲$: $N۲$ تشکيل شده است از همه حرکت هاي شدني $kempeswap$.

يك حرکت $kempeswap$ تعريف شده است به وسيله جابه جايي دو $kempechains$.

اگر ما تمرکز کنيم تنها روی درسها و برخورد آنها، به هرنمونه مساله را ميتوان به عنوان يک گراف G در نظر گرفت بطوري که راس هاي آن درس ها و يال هاي آن بين درس هايي است که دانش آموز ويا معلم مشترك داشته باشند. در يك زمانبندي شدني يك کمپ چين يك مجموعه است از راس هايي است که در زير گراف القا شده از دو دوره G ، به صورت يك مولفه به هم متصل اند. يك کمپ چين توليد مي کند يك تخصيص شدني جديد به وسيله جابجايي برچسب دوره هاي

شکل ۱: illustrations chain kempe

تخصیص یافته به درس هاي متعلق به يك يا دو كمپ چين مشخص. به عبارت ديگر فرض كنيد K_1 و K_2 دو كمپ چين در زير گراف متناظر با دو دوره t_i و t_j باشد، يك $kempeswap$ توليد مي كند يك تخصيص به وسيله جابجايي t_i با $(t_i \setminus (k_1 \cup k_2)) \cup (t_i \cap (k_1 \cup k_2))$ و جابجايي t_j با $(t_j \setminus (k_1 \cup k_2)) \cup (t_j \cap (k_1 \cup k_2))$. توجه كنيد كه در تعريف N_2 حداقل ۳ درس دخالت دارند. يعني $|K_1| + |K_2| \geq$ ۳ براي نمونه شكل (۱.a) به تصوير مي كشد يك زير گراف توليد شده به وسيله دو دوره t_i و t_j وجود دارد پنج كمپ چين:

$$K_a = \{c_1, c_2, c_7, c_8\}, K_b = \{c_3, c_6, c_9\}, K_c = \{c_4, c_{11}, c_{12}\}, K_u = \{c_5\}, K_e = \{c_{10}\}$$

در اين مثال هر اتاق در دوره t_i و t_j يک سخنرانی دارد . يک $kempeswap$ از K_a و K_c يک تخصيص جديد توليد می كند به وسيله حركت دادن $\{c_3, c_4, c_6\}$ به t_j و $\{c_9, c_{11,c_{12}}\}$ به t_i هماطور كه در شكل (۱.b) نشان داده شده است .

توجه كنيد كه در $kempeswap$ يكی از كمپ چين ها می تواند تهی باشد .

توجه كنيد كه تبادل دو كمپ چين می تواند به عنوان تعميمی از يک كمپ چين در نظر گرفت كه در متون ديگر بكار رفته است [۳]، [۱۴].

هنگامی كه دروس در دوره ها برنامه ريزی شدند تخصيص اتاقها ميتواندبه وسيله حل يک مساله $bipartite\ matching$ انجام شود [۱۲] . توجه كنيد كه در اينجا هم هيورستيک ها و هم الگوريتم های دقيق را می توان بكار گرفت ولی در اين مقاله [۹] ما برای تقويت اجرای الگوريتم خودمان از يک الگوريتم دقيق [۱۳] استفاده می كنيم كه در $o(|V|.|E|)$ اجرا می شود .

به منظور شدني ماندن جواب هاي همسايگي كمپ چين يك ويژگي مهم ديگري بايد برقرار باشد : تعداد درس هاي در هر دوره (بعد از تبادل كمپ چين) نبايد از اتاق هاي در دسترس تجاوز كند. براي مثال در شكل (۱) متناظر با تبادل يك كمپ چين ، تنها يك حركت شدني مي تواند توليد شود به وسيله تبادل درس ها در K_a در حالی كه تبادل هر يک از ۴ كمپ چين تنهای ديگر (K_b, K_c, K_d, K_e) نمی تواند جواب شدنی توليد كند زيرا اين حركت ها ويژگی را كه در بالا گفتيم را نقض می كنند بنابراين ممنوع می باشند. در واقع اين ويژگی محدوديت بزرگی را برای تعدادی از جوابهای

کاندید قابل پذیرش درتبادل یک کمپ چین به وجود می آورد. ما این محدودیت را نقض تخصیص اتاقها می نامیم هر چند به زودی با تبادل دوتایی کمپ چین که انجام شود نقض تخصیص اتاقها آزاد شده و یک تعدادی از حرکت های شدنی می تواند تولید شود. برای مثال در شکل (۱) تبادل ۳ دابل کمپ چین می تواند تولید شود به وسیله جابجایی

$$K_b \longleftrightarrow K_e \ , \ K_d \longleftrightarrow K_c \ , \ K_b \longleftrightarrow K_c$$

۷.۵ ارزیابی تدریجی و همسایگی کاهش :

به منظور ارزیابی همسایگی ها در یک روش کارآمد ما از یک تکنیک ارزیابی تدریجی استفاده می کنیم. ایده اصلی ماندن در یک ساختمان داده خاص و ارزیابی حرکت برای هر حرکت شدنی در جواب جاری.

هر بار یک حرکت انجام می شود و عناصر این ساختمان داده ها تحت تأثیر این حرکت بروز رسانی $(update)$ می شوند. با این حال ارزیابی حرکت از همسایگی پیشرفته $N_۲$ نیاز به تلاش محاسبات به مراتب بیشتر نبست به $N_۱$ به دلیل اجرای الگوریتم $matching$ دارد. به منظور صرفه جویی در زمان محاسباتی ما تلاش می کنیم برای استفاده از الگوریتم $matching$ تا حد ممکن کم.

بر طبق فرمول بندی مسأله هزینه های نرم می تواند دسته بندی شود به داخل هزینه های اتاق مربوطه $(S_۱, S_۲)$ و هزینه های دورهٔ مربوطه $(S_۳, S_۴)$.

از تعریف $N_۲$ واضح است که هزینه دورهٔ مربوطه $\triangle f_p$ می تواند محاسبه شود بدون فراخوانی الگوریتم $matching$ بنابراین محاسبات این قسمت ساده است اما محاسبه هزینه اتاق مربوطه $\triangle f_r$ وقت گیر است به دلیل هزینه محاسباتی بالاتر الگوریتم $matching$ در پیاده سازی ما، ما فقط بروز رسانی و ثبت می کنیم ارزش حرکت دورهٔ مربوطه $\triangle f_p$ برای جوابهای همسایهٔ $N_۲$ ، در حالی که برای ارزش حرکت اتاق مربوطه یک تکنیک کاهشی خاصی برای تصمیم گیری اینکه آیا الگوریتم $matching$ فراخوانی شود و یا نه بکارگرفته شده است. در حقیقت ما، استفاده می کنیم از هزینه (ارزش) دورهٔ مربوطه $\triangle f_p$ به عنوان برآورد خود حرکت کمپ. مخصوصاً اگر هزینه دورهٔ مربوطه $\triangle f_p$ امیدوار کننده باشد (یعنی $\triangle f_p \leqslant Z$ ، عملاً $Z = ۲$ تولید می کند نتایج رقابتی برای یک رده $(class)$ بزرگتر از نمونه ها)، آنگاه الگوریتم $matching$ را برای درست کردن تخصیص اتاق ها و بدست آوردن مجموع هزینه ارزیابی تدریجی $\triangle f$ ، فراخوانی می کنیم. در غیر این صورت جواب کاندید این همسایگی دور ریخته می شود. در این روش در هر تکرار فقط یک زیر مجموعه کوچکی از جوابهای همسایگی امیدوار کننده کاملاً ارزیابی می شوند، پس این امکان را به ما می دهد که

زمان زیادی صرفه جویی کنیم.

۸.۵ مدیریت لیست ممنوع :

درون TS یک لیست تابو معرفی شده است که ممنوع می کند بازبینی جوابهایی را که قبلاً دیده شده است. در الگوریتم TS ، زمانی که حرکت می دهیم یک سخنرانی (جلسه درس) را از یک مکان (هم اتاق و هم دوره) به جای دیگر (با استفاده از N_1) یا از یک دوره به دورهٔ دیگر (با استفاده از N_2) این سخنرانی نمی تواند یک حرکت رو به عقب به مکان قبلی (برای N_1) یا دوره قبلی (برای N_2) برای tt تکرار بعدی داشته باشد (tt نامیده می شود دورهٔ تابو). به طور دقیق تر در همسایگی N_1 ، اگر یک سخنرانی از یک درس C_i حرکت داده شود از یک مکان (t_j, r_k) به مکان دیگر آنگاه حرکت هر سخنرانی از درس C_i به مکان (t_j, r_k) اعلام شده است. و همینطور در همسایگی N_2 (هر حرکت تنها یا دوتایی کمپ چین) ، اگر یک سخنرانی از یک درس C_i از دورهٔ t_j به t_k جابجا شده باشد، تخصیص هر سخنرانی C_i به t_j با استفاده از یک (حرکت تنها یا دوتایی) حرکت کمپ چین تابو می باشد. دوره ممنوعیت $tt(C_i)$ از یک درس C_i به طور سازگار با کیفیت جواب جاری f و حرکات فرکانسی مربوط به سخنرانی(جلسه) درس C_i تنظیم شده است که به صورت زیر نشان داده می شود : $tt(C_i) = f + \varphi.q(C_i)$ که در آن φ پارامتری است که مقداری در بازهٔ $[0, 1]$ را اختیار می کند. قسمت اول از این تابع را می توان به این صورت توجیه کرد که یک جواب با ارزش جریمه های نرم بالا (زیاد)، باید یک دورهٔ ممنوعیت طولانی داشته باشد تابتواند از یک تله بهینه محلی فرار کند. ایده اصلی در پشت قسمت دوم برای جریمه کردن یک حرکتی است که خیلی تکرار شده است. ضریب φ که بصورت پویا مشخص می شود برابر است با نسبت تعداد برخوردهای درس C_i روی تعداد کل دروس و این نیز منطقی است که فکر کنیم حرکت دادن یک درس با یک تعداد زیادی برخورد، خطر بیشتری دارد، نسبت به درسی که دارای برخوردهای کمتری است.

۹.۵ معیار آرمانی و شرط توقف :

در الگوریتم TS ، حالت ممنوعیت از حرکت غیر فعال است اگر حرکت به حالت گذشته منجر به جوابی شود که از همهٔ جوابهایی که تا بحال بدست آمده بهتر باشد. و شرط توقف زمانی است که بهترین جواب نتواند بهبود یابد در طی حرکت Θ و ما به این عدد عمق TS می گوئیم.

۱۰.۵ ترکیب TS با آشفتگی :

TS و تکرار جستجوی محلی (ILS) دو متامبورسنیک شناخته شده ای هستند که کارآیی آنها برای حل به طور مستقل یک تعداد زیادی از مسائل ارضای محدودیت و بهینه سازی به اثبات رسیده است.

در این سمینار ما، امکان ترکیب آنها را برای رسیدن به یک عملکرد خیلی بالا برای مسأله CTT بررسی می کنیم. TS می تواند با زمان محاسبات طولانی و کوتاه استفاده شود. در حالت کلی زمان محاسبات طولانی به نتایج بهتر منجر می شود. با این حال اگر کل زمان محاسبات محدود باشد ترجیح داده می شود به ترکیب اجرای TS کوتاه با بعضی از عملگرهای تنوع قوی. جالب توجه است که ILS فراهم می کند مکانیزم متنوعی برای هدایت جستجو برای فرار از بهینه محلی جاری و حرکت به سمت مناطقی جدید امیدوار کننده در فضای جستجو [۷].

۱۱.۵ یک جریمه - استراتژی هدایت آشفتگی :

زمانی که بهترین جواب نمی تواند با استفاده بیشتر از TS بهبود یابد ما از یک اپراتور اغتشاش برای بازسازی جواب بهینه محلی بدست آمده استفاده می کنیم. قدرت آشفتگی یکی از مهمترین فاکتورهای در ILS است. به طور کلی از قدرت آشفتگی خیلی قوی باشد آن ممکن است عمل کند شبیه به یک شروع مجدد تصادفی. از طرف دیگر اگر آشفتگی خیلی ضعیف باشد پس از جستجو در یک فضایی که محدود شده به یک منطقه کوچک دوباره به بهینه محلی قبلی سقوط می کند به منظور هدایت کارآمد جستجو برای حرکت به سمت مناطق امیدوار کننده جدید از فضای جستجو، بکار می گیریم یک جریمه که هدایت کند اپراتور آشفتگی را برای ویرانی جواب بهینه محلی بدست آمده. این اپراتور براساس شناسایی یک مجموعه q عضوی از اولین سخنرانی هایی که جریمه به شدت بالایی دارند و انتخاب تصادفی از یک تعداد حرکت همسایگی داده شده. $(q = ۳۰)$

تعداد کل حرکات های اغتشاشی را قدرت اغتشاش می نامیم و با η نمایش می دهیم. مخصوصاً زمانی که فاز TS جاری پایان می پذیرد همهٔ سخنرانی ها به صورت غیر افزایش براساس جریمه دخیل در محدودیت نرمشان رتبه بندی می شوند. آنگاه η سخنرانی انتخاب می شوند از q تای اول که جریمه بالاتری دارند جاهایی که رتبه سخنرانی k ام انتخاب می شود براساس توزیع احتمال زیر: $P(k) \propto k^{-\varphi}$ که در آن φ یک عدد حقیقی مثبت (به طور تجربی ۴). بعد از آن η حرکت شدنی از $simpleswap$ یا $kempeswap$ به صورت تصادفی و متوالی انجام می شود که هر یک شامل حداقل یکی از η سخنرانی انتخاب شده می باشد.

همانطور که قبلاً ذکر شده قدرت آشفتگی η یکی از مهمترین اجراء ILS است که تعیین می کند کیفیت شکاف بین دو جواب قبل و بعد از آشفتگی را. در مورد ما، η به طور سازگار تنظیم می شود و مقداری در بازهٔ $[\eta_{min}, \eta_{max}]$ می گیرد (به طور تجربی $\eta_{min} = ۴$ و $\eta_{max} = ۱۵$).

الگوریتم ATS ، فاز تشدید (TS) و تنوع (اغتشاش ILS) را ادغام می کند. بجای ترکیب ساده الگوریتم های (TS) و ILS ما تلاش می کنیم که آنها را به یک روش معنی دار با هم ادغام کنیم. عمق TS (Θ) و قدرت آشفتگی (η) به نظر می رسد که دو پارامتر ضروری برای کنترل رفتار ATS هستند. از یک جهت یک مقدار بزرگتر Θ یک جستجوی شدیدتر (فشرده تر، متمرکزتر) را تضمین می کند از سوی دیگر یک η بزرگتر متناظر است با امکان بیشتر برای فرار از جواب بهینه محلی جاری. به منظور رسیدن به یک تعویض پیوسته میان تشدید و تنوع، ما یک مکانیسمی را طراحی می کنیم که این دو پارامتر مهم را بر طبق نتایج سابق پروسه جستجو به طور پویا و سازگار تنظیم می کند.

۱۲.۵ الگوریتم $Adoptive\ Tobu\ search$ (ATS) :

در شروع جستجو ما اجرا می کنیم یکی TS کوتاه جاهایی که عمق TS کوچک است $(\Theta = ۱۰)$ ؛ موقعی که TS نمی تواند بهبود دهد بهترین جواب را، اغتشاش در بهترین جواب با یک قدرت ضعیف $\eta = \eta_{min}$ بکار گرفته می شود. در هنگام پروسه جستجو، ما ثبت می کنیم تعدادی از فازهای TS (نمایش داده شده با ξ) بدون بهبود در هزینه تابع. عمق TS (Θ) و قدرت اغتشاش η به طور پویا به صورت زیر تنظیم می شوند: زمانی که مینیمم محلی بدست آمده توسط TS امیدوار کننده است یعنی آن به بهترین جواب جاری نزدیک است $(f \leqslant f_{best} + ۲)$, Θ به تدریج افزایش می یابد برای اطمینان از یک جستجوی شدیدتر (فشرده تر) تا زمانی که هیچ بهبودی امکان پذیر نباشد یعنی $\Theta = (۱ + \mu)\Theta$ در هر تکرار $(\mu = ۰/۶)$.

به طور مشابه قدرت اغتشاش به طور تدریجی افزایش می یابد ، زیرا تنوع بیشتر قوی می کند جستجو را اگر (نسبت به اینکه) فازهای غیر بهبودی TS افزایش یابد.

علاوه بر این جستجو بر می گردد به TS کوتاه بعد از هر اغتشاش در حالی که قدرت اغتشاش بر می گردد به η_{min} به محض اینکه یک جواب بهتر یافت شود. برای معیار پذیرش در پروسه اغتشاش ما از یک تکنیک قوی استفاده می کنیم یعنی تنها جواب بهتر پذیرفته می شود به عنوان بهترین جواب جاری . به محض اینکه جواب بهینه محلی $X^{*'}$ که توسط TS بدست آمده از بهترین جواب X^* که تاکنون یافت شده است بهتر باشد ما جایگزین می کنیم بهترین جواب

شکل ۲: $AdaptiveTabuSearch$

شناخته شده X^* را با $X^{*'}$ همانطورکه در خط ۱۸ و ۱۹ الگوریتم نشان داده شده است.

در این مقاله [۹] ما از دو شرط توقف همانطورکه توصیف شد استفاده می کنیم:

۱. محدودیت زمانی

۲. یک تعداد ماکزیمم حرکت ها

۶ الگوریتم ژنتیک

زمانبندی کلاسها و تتصیص اتاقها کار اصلی برای حل یک مساله زمانبندی محسوب می شود از آنجایی که این دو کار به شدت به هم وابسته اند کامل کردن آنها به طور جداگانه کار سختی است با وجود این کامل کردن این کارها به طور همزمان نا کارآمد بنظر می رسد .به عنوان یک روش برای حل این مشکل ما یک روشی که ژنتیک (GA) را در دو گام استفاده می کند ، ارائه می کنیم و آنرا TGA [۳] مینامیم.

در الگوریتم TGA ما از دو نوع جمعیت استفاده میکنیم . جمعیت اول در ارتباط با زمانبندی کلاسها وجمعیت دوم برای تخصیص اتاقهاست . این جمعیتها بطور مستقل تکامل می یابند و یک مقدار هزینه ای برای هر فرد محاسبه می شود سپس به منظور محاسبه $fitness$ تعدادی از کم هزینه ترین افراد از هر جمعیت انتخاب شده و این افراد باهم ترکیب می شوند .

۱.۶ محدودیت های مساله

$H_۱$: هیچ استادی بیش از یک بار دریک دوره ظاهر نشود .

$H_۲$: درسهای چند کلاسه باید به دوره های مشابه ای تخصیص داده شوند .

$H_۳$: درسهای چند دوره ای باید از دوره های متوالی استفاده کنند .

$H_۴$: برای هر درس باید یک اتاق با اندازه و نوع مناسب تخصیص داده شود .

$H_۵$: هیچ اتاقی بیش از یک بار در هر دوره ظاهر نشود .

علاوه بر محدودیتهای پایه ای ما چند محدودیت دیگر را نیز در نظر می گیریم.

$S_۱$: بعضی از استادهای باید در برخی از دوره ها ظاهر نشوند.

$S_۲$: بعضی از اتاقها در برخی از دوره های خاص نباید استفاده شوند.

$S_۳$: برای بعضی از گروهها هیچ درسی در یک دوره خاص ظاهر نشود.

$S_۴$: بعضی از درسها باید تدریس شوند بلافاصله بعد از تدریس یک درس خاص .

[۳] [۱۵]

شکل ۳: $The genotypes for TGA$

۲.۶ الگوریتم TGA

در TGA ما از دو نوع ژنوتیپ استفاده می کنیم . شکل (۳) [۴] نشان می دهد ژنوتیپ ها را، یکی از ژنوتیپ ها برای زمانبندی کلاسها و دیگری برای تخصیص اتاقها .

ژنوتیپی که برای زمانبندی کلاسها استفاده می شود شامل یک جدول زمانی برای هر کلاس ؛ و هر $timeslot$ در یک جدول زمانی شامل اطلاعات دروس می باشد. اطلاعات دروس شامل ID (شماره شناسایی) درس و ID استاد می باشد. ژنوتیپ برای تخصیص اتاقها شامل ID هر اتاق است. در شکل ۴ فلوچارت TGA نشان داده شده است. در فاز اول، عملگرهای ژنتیک برای زمانبندی کلاسها روی جمعیت اعمال می شوند و هزینه مربوط به محدودیتها برای زمانبندی کلاسها محاسبه می شود. در فاز دوم عملگرهای ژنتیک برای تخصیص اتاقها بر روی جمعیت اعمال می شوند. و در نهایت، ما ترکیب می کنیم این دو نوع از افراد را و مقدار $fitness$ را محاسبه می کنیم. این مراحل تکرار می شوند تا زمانی که شرط توقف برآورده شود. بعداً، ما جزئیات هر عملگر ژنتیک را توضیح می دهیم. فاز زمانبندی کلاس: در اینجا ما عملگرهای ژنتیک را برای زمانبندی کلاس توضیح می دهیم.

[۴] [۱۶]

شکل ۴: $AowdiagramofTGA$

شکل ۵: $Crossoverforthegenotyperelatedtoclassscheduling$

۳.۶ انتخاب و تولید مثل

استراتژی ما استفاده از چرخ رولت و حفظ نخبگان است. یک فرد با هزینه کم به احتمال زیاد در مرحله $selection$ ، انتخاب خواهد شد. نخبه یک جفت از دو نوع از افراد است. در اینجا ما نشان می دهیم یک شخص را برای زمانبندی کلاسها با C_i و یک شخص برای تخصیص اتاقها را با R_j. و یک جفت ازافراد را با $\langle C_i, R_j\rangle$ نمایش می دهیم. زمانی که مجموع هزینه ها برای $\langle C_i, R_j\rangle$ در محاسبه $fitness$ مینیمم شود با این جفت به عنوان یک نخبه رفتار می شود یعنی عملگردهای ژنتیک مانند تقاطع و جهش بر روی C_i و R_j اعمال نمی شوند.

۴.۶ تقاطع

ما استفاده می کنیم از تقاطع یک نقطه ای و نقاط تقاطع به مرزهای بین جدول زمانی برای هر کلاس محدود شده اند. در شکل (۵) یک مثال از عملگرد تقاطع نشان داده شده است. در این شکل یک نقطه تقاطع مرز بین کلاس ۲ و کلاس ۳ می باشد. بعد از تقاطع همهٔ دروس باید دقیقاً یکبار ظاهر شوند به دلیل این محدودیت، به نظر می رسد که عملگر تقاطع نتواند یک جدول زمانبندی را تغییر دهد. با این حال، این مشکل با بکارگیری یک عملگر تعمیر حل شده است. با بکارگیری عملگر تقاطع بسیاری از شرایط (تمایلات) دروس چندکلاسه نقض می شود عملگر تعمیر جدول زمانبندی را به منظور تعمیر نقض دروس چند کلاسه اصلاح می کند . پس عملگر تقاطع نیز منجر به تغییر برنامه هر کلاس می شود.

۵.۶ جهش

در جهش، اطلاعات یک درس از یک بازهٔ زمانی خاص جابجا می شود با اطلاعات یک درس که بطور تصادفی از یک $timeslot$ انتخاب شده است. اگر درس آن $timeslot$ درس چند دوره ای باشد آنگاه اطلاعات درس در چندین $timeslot$ تغییر می کند شکل (۶) یک مثال از جهش را نشان داده است.

شکل ۶: $Mutation for the genotype related to class scheduling$

شکل ۷: $An example of a timetable before the repair operation$

۶.۶ تعمیر

بعد از تقاطع و جهش چندین درس چند کلاسه ممکن است نقض شوند. عملگر تعمیر تلاش می کند این درسها را به وسیله حرکت دادن دروس نقض شده در صورت امکان به $timeslot$ های مختلف تعمیر کند. زمانی که ما نمی توانیم تعمیر کنیم نقض های یک درس را یک جریمه داده می شود به فردی که به خانه نقض شده درس متعلق است. ما توضیح می دهیم مقدار جریمه را زمانی که محاسبه هزینه در هر گام را مطرح کنیم. عملگر تعمیر نمی کند نقض دروس را موقعی که تعدادی از دروس چند کلاسه یا دروس چند دوره ای به تازگی نقض شده باشند به وسیله تعمیر نقض دروس. در روش ما، دروس با پیچیدگی بیشتر زودتر تعمیر می شوند نسبت به دروس با پیچیدگی کمتر. در اینجا ما پیچیدگی یک درس را به صورت $PC + c$ تعریف می کنیم به طوریکه در آن P تعداد دوره های مورد نیاز برای آن درس و C تعداد کلاسها برای مسأله زمانبندی و c تعداد کلاسهایی است که آن درس در آن تدریس می شود. با توجه به تعریف، دروس چند دوره ای که به دوره های زیادی نیاز دارند پیچیده ترین دروس در نظر گرفته می شوند و دروس چند کلاسه ای که برای تعداد زیادی از دانشجویان (کلاسها) تدرس می شوند دروس با پیچیدگی بیشتردر نظر گرفته می شوند. درشکل ۷ درس ۲ پیچیده تر از درس ۱ می باشد و پیچیدگی آنها به ترتیب ۱۰ و ۷ می باشد برای درس ۱ فقط به یک دوره نیاز داریم ولی ۳ جلسه در هفته $c = ۳$ و چون تعداد کلاسهای جدول ۴ تا است پس $C = ۴$ یعنی برای درس ۱ داریم:

$$۱ \times ۴ + ۳ = ۷$$

برای توضیح رفتار عملگر تعمیر، ما از این مثال استفاده می کنیم.

شکل ۸: $An example of a timetable after the repair operation$

در شکل ۷ ، دروس چند کلاسه ۱ و ۲ نقض شده اند از آنجایی که درس ۲ از درس ۱ پیچیده تر است ابتدا برای تعمیر درس ۲ تلاش می شود. عملگر تعمیر تلاش می کند که درس ۲ را در کلاس $B-1$ به دوره های ۳ و ۴ حرکت دهد و یا درس ۲ را در کلاس $D-1$ به دوره های ۱ و ۲ منتقل کند هنگامی که درس ۲ در کلاس $B-1$ به دوره های ۳ و ۴ منتقل می شود، درس ۷ به تازگی نقض شده پس درس ۲ در کلاس $D-1$ به دوره های ۱ و ۲ منتقل می شود به روش مشابه، درس ۱ در کلاس $A-1$ حرکت می کند به دوره ۳.

در شکل ۸ زمانبندی را بعد از عملگر تعمیر نشان داده شده است به طوریکه هیچ درس چند کلاسه ای نقض نشده است. از آنجایی که عملگر تعمیر یک عملگر ساده ای است هزینه محاسبات بالایی ندارد. با این حال بسیاری از دروس نقض شده را می توان با این عملگر تعمیر کرد.

۷.۶ محاسبه هزینه

در فاز برنامه ریزی کلاس، فقط محدودیتهای زمانبندی کلاسها، برای محاسبه هزینه استفاده می شوند. در معادله (۲) هزینه برنامه ریزی C_i با جمع مقادیر جریمه تعریف شده است.

$$Scheduling\ Cost(C_i) = \sum_{k \in all\ constraints\ for\ class\ scheduling} Penalty\ value(k) \qquad (۲)$$

هنگامی که دروس چند دوره ای و دروس چند کلاسه نقض شوند مقدار جریمه ارزیابی می شود، مقدار جریمه بر طبق وسعت نقض آنها ارزیابی می شود. اکنون ما فرض می کنیم که درس $T-$ برای n کلاس تدریس می شود و این درس به m دوره متوالی نیاز دارد. علاوه بر این ما فرض می کنیم که این درس از P نوع (Pn) از $timeslots$ جدول زمانبندی استفاده می کند یک مقدار جریمه برای نقض این درس به m و P وابسته است و مقدار آن را با $(p-۱)m$ تعریف می کنیم. درشکل (۷) m و P برای درس ۱ به ترتیب برابر با ۱ و ۳ است پس مقدار جریمه برای نقض این درس به صورت $(۳-۱) \times ۱ = ۲$ محاسبه می شود. به روش مشابه مقدار جریمه برای درس ۲ در همان جدول به صورت $(۱-۲) \times ۲ = ۲$ محاسبه می شود زیرا درس ۲ از دو نوع $timeslots$ استفاده می کند یعنی [دوره ۱ و دوره ۳] و [دوره ۳ و دوره ۴].

۸.۶ فاز تخصیص اتاق

در فاز تخصیص اتاق، ما استفاده می کنیم از عملگرهای ژنتیک ای مشابه با آنچه که در فاز برنامه ریزی کلاسها استفاده کردیم به جز برای جهش. پس ما اینجا فقط عملگر جهش را توضیح می دهیم. در جهش ما، یک اتاق را برای یک درس

شکل ۹: $Mutation for the genotype related to room allocation$

جهش یافته جابجا می کنیم. شکل (۹) یک جهش برای درس T را نشان می دهد. در موقعی که اتاق تخصیص می یابد به یک درسی که جهش یافته است، ممکن است یک اتاقی که اندازه اش متفاوت است با اندازه اتاق تخصیص جاری با این درس جابجا شود ، اما اتاقی با یک نوع متفاوت به آن تخصیص نمی یابد. برای مثال، ما فرض می کنیم که درس T در شکل (۹) نیاز به اتاق سخنرانی با اندازهٔ متوسط دارد. موقعی که یک اتاق تخصیص می یابد برای درس T که جهش یافته است ما ممکن است که یک اتاق سخنرانی که اندازهٔ آن متوسط نباشد را به آن تخصیص دهیم که قبلاً اندازهٔ آن بزرگ و یا کوچک باشد اما یک اتاق کامپیوتر را به آن تخصیص نمی دهیم.

۹.۶ فاز محاسبه $fitness$

در مرحله $fitness$ ، مقدار $fitness$ محاسبه می شود بعد از آنکه برخورد اتاق بررسی شده باشد. در بررسی برخورد اتاق، n شخص با کمترین هزینه از دو نوع جمعیت انتخاب می شوند و افراد انتخاب شده ترکیب می شوند. هر فردی که از جمعیت برنامه ریزی کلاسها انتخاب شود با همهٔ افراد انتخاب شده از جمعیت تخصیصی اتاقها ترکیب می شود پس n^2 جفت از افراد ساخته می شوند. برای هر جفت ما بررسی می کنیم که آیا برخورد اتاق اتفاق افتاده است و یا نه و یک هزینه برخورد اتاق محاسبه می شود در اینجا ما نمایش می دهیم هزینه برخورد اتاق برای جفت افراد $\langle C_i, R_j \rangle$ را به صورت $Room\ clashing\ cost(C_i, R_j)$ و ما فرض می کنیم که $Room\ clashing\ cost(C_i, R_j)$ متناسب با تعداد برخوردهای اتاق است. با استفاده از $Room\ clashing\ cost(C_i, R_j)$ ما مقدار هزینه همه افراد هر دو نوع جمیعت را به روز می کنیم . معادله های (۳) و (۴) نشان می دهد تابع هایی را که مقدار هزینه افراد وابسته به برنامه ریزی کلاسی را بروز رسانی می کنند که در آن $Class\ Cost(C_i)$ مقدار هزینه بروز شده C_i است. زمانی که C_i انتخاب شده است در بررسی برخورد اتاق، ما از معادله (۳) برای محاسبه $Class\ Cost(C_i)$ استفاده می کنیم.که در آن $Class\ Cost(C_i)$ برابر است با مینیمم مقدار هزینه برخورد اتاق بعلاوه $Scheduling\ Cost(C_i)$ هنگامی که C_i انتخاب نشود در هنگام بررسی برخورد اتاق، $Class\ Cost(C_i)$ توسط فرمول (۴) محاسبه می شود در (۴)، میانگین مقدار مینیمم هزینه های برخورد اتاق یک جریمه ای برای انتخاب نشدن در مرحله چک کردن برخورد اتاقها

شکل ۱۰: $Overview of the room clash check$

است. با استفاده از هزینه های به روز شده $Class\ Cost(C_i)$ مقدار $fitness$, $Class\ fitness(C_i)$ با استفاده از معادله (۵) محاسبه می شود.

$$Class\ Cost(C_i) = \min_j (Room\ Clashing\ Cost(C_i, R_j)) + Scheduling\ Cost(C_i) \quad (۳)$$

$$Class\ Cost(C_i) = \underset{k}{average} \left(\min_j (Room\ clashing\ Cost(C_k, R_j)) \right) + Scheduling\ Cost(C_i) \quad (۴)$$

$$Class\ fitness(C_i) = \max_k (Class\ Cost(C_k) - Class\ Cost(C_i) \quad (۵)$$

مقدار $fitness$, اعضای تخصیص اتاقها نیز به طریق مشابه محاسبه می شود . مرحله انتخاب و تولید مثل با بکار گیری مقدار $fitness$, اجرا می شود .

مراجع

[1] Sanja Petrovic and Edmund Burke : Handbook of Scheduling Algorithms Models and Performance Analysis; Chapter : University Timetabling. 7

[2] A. SCHAERF , A Survey of Automated Timetabling , Dipartimento di Ingegneria Elettria Gestionale e Meccanica, Universit‘ a di Udine, Via delle Scienze 208, 33100 Udine, Italy ,Artificial Intelligence Review 13: 87–127, 1999. 7

[3] M. Chiarandini, M. Birattari, K. Socha, O. Rossi-Doria, An effective hybrid algorithm for university course timetabling, Journal of Scheduling 9 (2006) 403–432. 5, 16

[4] R.V. Rasmussen, M.A. Trick, Round robin scheduling – a survey, European Journal of Operational Research 188 (3) (2008) 617–636. 5

[5] S.M. Al-Yakoob, H.D. Sherali, Mathematical programming models and algorithms for a classfaculty assignment problem, European Journal of Operational Research 173 (2) (2006) 488–507. 5

[6] M.M. Rodrigues, C.C. de Souza, A.V. Moura, Vehicle and crew scheduling for urban bus lines, European Journal of Operational Research 170 (3) (2006) 844– 862. 5

[7] H.R. Lourenco, O. Martin, T. Stützle, Iterated local search, in: F. Glover, G. Kochenberger (Eds.), Handbook of Meta-heuristics, Springer-Verlag, 2003, pp. 321–353. 19

[8] Cooper T., Kingston J., The Complexity of Timetable Construction Problems, Lecture Notes in Computer Science, Vol. 1153, 1996,pp. 281-295.

[9] Zhipeng Lü , Jin-Kao Hao : Adaptive Tabu Search for course timetabling , European Journal of Operational Research 200 (2010) 235–244. 11, 16, 21

[10] F. Glover, M. Laguna, Tabu Search, Kluwer Academic, Boston, 1997. 14

[11] L. Di Glover, A. Schaerf, Neighborhood portfolio approach for local search applied to timetabling problems, Journal of Mathematical Modeling and Algorithms 5 (1) (2006) 65–89. 14

[12] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall Inc., 1982. 16

[13] J.C. Setubal, Sequential and Parallel Experimental Results with Bipartite Matching Algorithms, Technical Report EC-96-09, Institute of Computing, University of Campinas, Brasil, 1996. 16

[14] J. Thompson, K. Dowsland, A robust simulated annealing based examination timetabling system, Computer and Operations Research 25 (1998) 637–648 . 16

[15] Edmund Burke Wilhelm Erben , Practice and Theory of Automated Timetabling III, Third International Conference, PATAT 2000 Konstanz, Germany, August , Selected Papers ,Chapter4 : A Co-evolving Timeslot/Room Assignment Genetic Algorithm Technique for University Timetabling 22

[۱۶] سید امیر حسن منجمی ، سولماز مسعودیان، افسانه استکی، ناصر نعمت بخش:طراحی جدول زمانبندی خودکار برای دروس دانشگاهی با استفاده از الگوریتم های ژنتیک، نشریه علمی پژوهشی فناوری آموزش ، سال چهارم ، جلد ۴ ، شماره ۲ ، زمستان ۱۳۸۸ ۲۳