به نام خدا

پروژه ۱

محمد علی حیدری- ۹۸۱۲۰۱۸

محمد نورمحمدی - ۹۸۱۲۰۵۴

کوروش مسلمی - ۹۸۱۳۰۲۷

در این گزارش سیر مدل های آزمایش شده تا رسیدن به مدل نهایی بررسی می شود. این مدل ها اشتراکات زیادی در متغیر ها و پارامتر ها دارند بنابراین برای طولانی نشدن گزارش همه آن ها یکجا معرفی شده اند. کد گمز و پایتون همه مدل هایی که اشاره می شوند ضمیمه این فایل خواهد شد. در پیاده سازی های انجام شده، گمز اطلاعاتی را از اکسل نمی خواند بلکه اطلاعات توسط پایتون به کمک کتابخانه های pandas و numpy خوانده شده و پس از تبدیل شدن به فرمت مناسب در یک شی از کلاس GamsDatabase ذخیره می شوند. سپس از طریق api گمز در پایتون این اطلاعات به گمز فرستاده می شود.

متغیر های تصمیم:

$q_{i,t}$: متغیر باینری که اگر امتحان درس i در روز t برگزار شود.

$c_{j,t}$: متغیر صحیح و نامنفی که نشان دهنده تعداد امتحان های دانشجوی j در روز t است.

$v_{j,t}$: متغیر صحیح و نامنفی که نشان می دهد دانشجوی j در روز t چند امتحان دیگر اضافه بر ۱ امتحان داشته است.(برای آزمایش دوم)

$c'_{j,w}$: متغیر صحیح و نامنفی که نشان دهنده تعداد امتحان های دانشجوی j در هفته w است.

u_j: متغیر صحیح و نامنفی که نشان دهنده ماکسیمم تعداد امتحان های دانشجو j در یک هفته در بازه امتحانات است.

پارامتر ها:

$a_{j,i}$: پارامتر باینری که آیا دانشجوی j درس i را دارد یا خیر.

$p_{i,t}$: پارامتر باینری که آیا درس i در روز t برگزار میشود یا خیر.

w_0: پارامتر صحیح و نامنفی که نشان می دهد بازه برنامه ریزی برای امتحانات چند هفته ای است.

مقادیر اندیس ها با توجه به داده های ترم جاری:

w: اندیس هفته ها از ۱ تا w_0 می باشد.

i: اندیس درس ها از ۰ تا ۵۶ می باشد.

j: اندیس دانشجو ها از ۰ تا ۵۷۹ می باشد.

t وt′: اندیس روز ها از ۰ تا $7w_0 - 1$ می باشد.

فرمول بندی شماره ۱:

$$\min \quad z=\sum_{j=0}^{579}\sum_{t=0}^{13}\sum_{t^{'}=0:t\geq t^{'}}^{13}\frac{c(j,t)\times c(j,t^{'})}{e^{t-t^{'}}}$$
$$s.t.$$
$$\sum_{t=0:p(i,t)=1}^{13} q(i,t)=1 \qquad \forall i=0,1,...,56$$
$$c(j,t)-\sum_{i=0:a(j,i)=1}^{56} q(i,t)=0 \qquad \forall j=0,1,...,579, \forall t=0,...,13$$

در این فرمول بندی تابع هدف برای هر دو امتحان نزدیک به هم یک دانشجو جریمه تخصیص می دهد. قید اول تضمین می کند در بازه دو هفته ای امتحانات حتما در یکی از روز هایی که درس i برگزار می شود امتحان گرفته شود و امتحانی در زمانی غیر از زمان آن درس برگزار نشود. قید دوم نیز به c(j,t) مقدار می دهد تا بعدا در تابع هدف استفاده شود.

مشاهداتی از فرمول بندی شماره ۱:

با توجه به اینکه تابع هدف این فرمول بندی غیرخطی بود مسئله به صورت MINLP در گمز با بهینه یاب Baron حل شد.(فایل project1_v1.2.gms) در این فرمول بندی محدودیت حداکثر یک امتحان در یک روز برای دانشجو به صورت soft لحاظ شده بود اما پس از اجرا برای بازه دو هفته ای به مدت ۷ ساعت و رسیدن به یک integer solution مشاهده شد برخی دانشجو ها وجود دارند که در یک روز بیش از یک امتحان دارند.(فایل outputs/t0.lst)

در این گام دو آزمایش انجام شد. در آزمایش اول تابع هدف برابر صفر قرار داده شد و قید های اول و دوم به صورت قبل حفظ شدند و قید حداکثر یک امتحان در یک روز نیز اضافه شد(فایل project1_experiment1.gms). در این آزمایش مشخص شد برای بازه دو هفته ای و حتی سه هفته ای جواب شدنی ای که شرط حداکثر یک امتحان در یک روز برای دانشجو را برقرار کند، وجود ندارد.

آزمایش دوم برای شناسایی دانشجو هایی بود که باعث می شوند این جواب شدنی وجود نداشته باشد.

$$\min \quad z=\sum_{j=0}^{579}\sum_{t=0}^{13} v_{j,t}$$
$$s.t.$$
$$\sum_{t=0:p(i,t)=1}^{13} q(i,t)=1 \qquad \forall i=0,1,...,56$$
$$\sum_{i=0:a(j,i)=1}^{56} q(i,t)-v(j,t)\leq 1 \qquad \forall j=0,1,...,579, \forall t=0,...,13$$

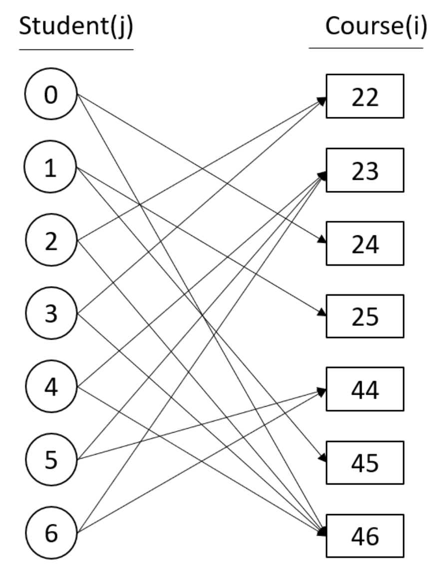
در این بررسی ۱۱۸ دانشجو شناسایی شدند که شیوه انتخاب دروس آن ها باعث نشدنی شدن مسئله در دو هفته می شدند. برای اطمینان بیشتر پس از حذف این ۱۱۸ دانشجو پاسخ فرمول بندی ۱ شرایط مسئله صدق می کرد. لیست این دانشجویان در فایل

problematic_students.txt موجود است و با فعال کردن متغیر باینری exclude_problematic_students در کد پایتون می توان مدل را بدون اسامی این افراد حل کرد و شرایط آزمایش را باز تولید کرد.(فایل project1_experiment2.gms)

پس از مطمئن شدن از نشدنی بودن جواب برای دو و سه هفته سعی شد برای چهارهفته فرمولبندی شماره ۱ با رعایت حداکثر یک امتحان در یک روز برای دانشجو اجرا شود.(فایل project1_v.1.3.gms) اما متاسفانه به دلیل پیچیده شدن بیش از حد تابع هدف و زیاد شدن دوتایی های t و t' به جوابی در زمان معقول(حداقل ۷ ساعت) نرسیدیم و در پایان هفت ساعت no solution returned دریافت کردیم. حتی سعی شد با استفاده از شرایط خاص داده های مسئله که در آخر هفته امتحانی برگزار نمی شود تعداد زوج های t و t' کمتر شود (فایل project1_v.1.4.gms) اما همچنان پاسخی در زمان معقول دریافت نشد. (فایل outputs/t1.lst)

تحلیل جواب فرمول بندی ۱ برای داده های محدود:

در این بررسی تنها برنامه ریزی امتحانات ۶ دانشجو را به مدل می سپاریم و جواب مدل را بررسی می کنیم:



t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
day	ش	ی	د	س	چ	پ	ج	ش	ی	د	س	چ	پ	ج
	44		44					44		44				
	45		45					45		45				
	46		46					46		46				
	22		24					22		24				
	23		25					23		25				

در شکل فوق اعداد زیرجدول تمام زمان هایی است که امتحان یک کد درس خاص می توانست برگزار شود و اعداد سبز رنگ زمان انتخاب شده توسط مدل برای برگزاری امتحان می باشد که اگر این به ازای هر دانشجو امتحان ها را بررسی کنیم کاملا مشخص است مدل سعی کرده است به درستی فاصله بین امتحانات را تا حد امکان زیاد کرده و از برگزاری امتحان های زیاد در یک روز جلوگیری کند.

فرمول بندی شماره ۲:

$$\min \quad z=\sum_{j=0}^{579} u_j$$

$$s.t.$$

$$\sum_{t=0:p(i,t)=1}^{13} q(i,t)=1 \qquad \forall i=0,1,\ldots,56$$

$$c(j,t)-\sum_{i=0:a(j,i)=1}^{56} q(i,t)=0 \qquad \forall j=0,1,\ldots,579, \forall t=0,\ldots,13$$

$$c(j,t)\le 1 \qquad \forall j=0,1,\ldots,579, \forall t=0,\ldots,13$$

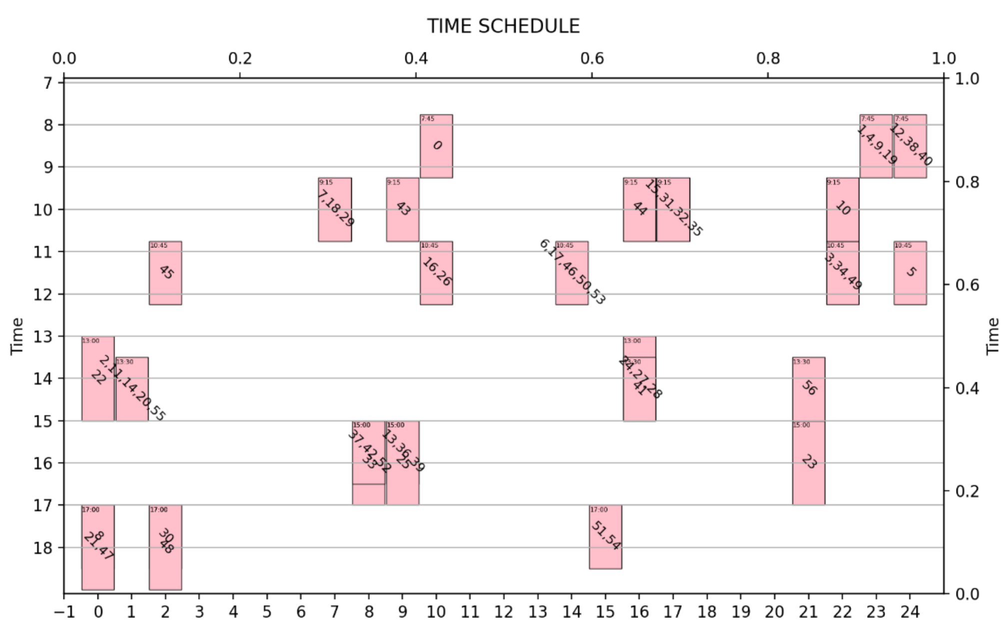
$$c^{'}(j,w)-\sum_{t=7(w_0-1)}^{7w_0-4} c(j,t)=0 \qquad \forall j=0,1,\ldots,579, \forall w=1,\ldots,w_0$$

$$u(j)\ge c^{'}(j,w) \qquad \forall j=0,1,\ldots,579, \forall w=1,\ldots,w_0$$

در این فرمول بندی با توجه به اینکه در داده ها امتحانی در آخر هفته برگزار نمی شود و همچنین طبق شرایط مسئله در هر روز دانشجو یک امتحان می دهد نتیجه می شود در هفته حداکثر چهار امتحان برای دانشجو قابل تعریف است. مدل فوق سعی می کند بیشینه امتحان هر دانشجو را در بین هفته های بازه امتحانات، کمینه کند. اگرچه این هدف غیرخطی است ولی می توان آن را به صورت فوق خطی کرد. یک ایراد این تابع هدف این است که بین فاصله امتحانات در یک هفته بهینه سازی انجام نمی دهد و فقط روی تعداد کار می کند ولی ایده کلی این است: *برای اینکه ماکسیمم امتحانات یک دانشجو در بین هفته ها کم شود پاسخ به سمت پخش امتحانات آن دانشجو در بین هفته ها می رود.*

این مدل را برای بازه ۴ هفته ای به مدت ۶ ساعت با بهینه یاب CPLEX حل کردیم به Integer Solution رسیدیم.

(فایل project1_v1.5.gms) برنامه ریزی انجام شده توسط این مدل به صورت زیر است: (فایل مکمل در پوشه outputs/t3)



فرمول بندی شماره ۳:

$$\min \quad z=\sum_{j=0}^{579}\sum_{t=0}^{13}\sum_{t'=0:t<t'}^{13}-c(j,t)\times c(j,t')\times(t'-t)^2+50\sum_{j=0}^{579}u_j$$

$$s.t.$$

$$\sum_{t=0:p(i,t)=1}^{13} q(i,t)=1 \qquad \forall i=0,1,\dots,56$$

$$\sum_{t=0:p(i,t)=0}^{13} q(i,t)=0 \qquad \forall i=0,1,\dots,56$$

$$c(j,t)-\sum_{i=0:a(j,i)=1}^{56} q(i,t)=0 \qquad \forall j=0,1,\dots,579, \forall t=0,\dots,13$$

$$c(j,t)\le 1 \qquad \forall j=0,1,\dots,579, \forall t=0,\dots,13$$

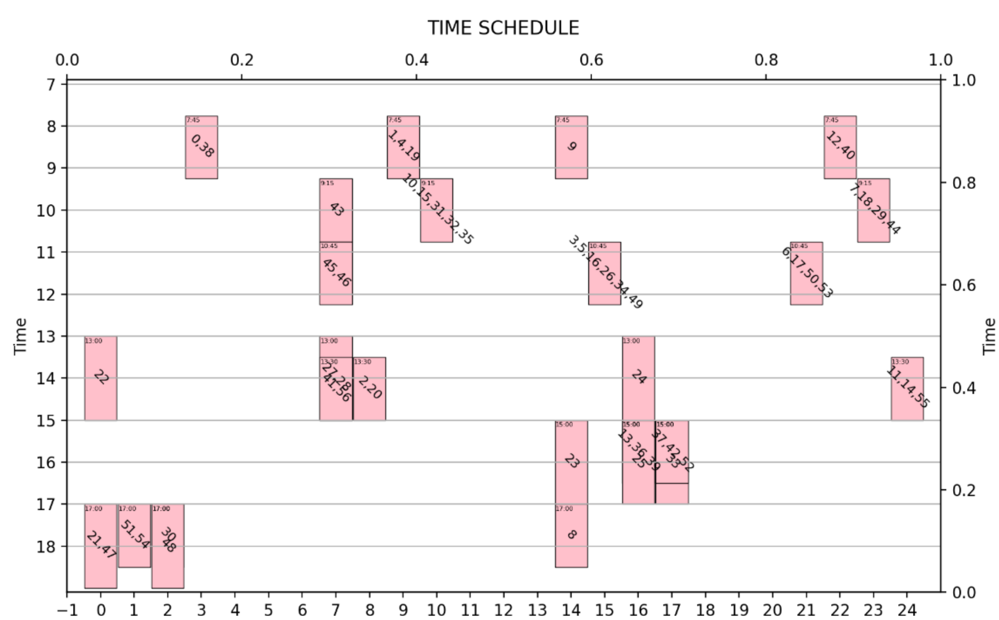
$$c'(j,w)-\sum_{t=7(w_0-1)}^{7w_0-4} c(j,t)=0 \qquad \forall j=0,1,\dots,579, \forall w=1,\dots,w_0$$

$$u(j)\ge c'(j,w) \qquad \forall j=0,1,\dots,579, \forall w=1,\dots,w_0$$

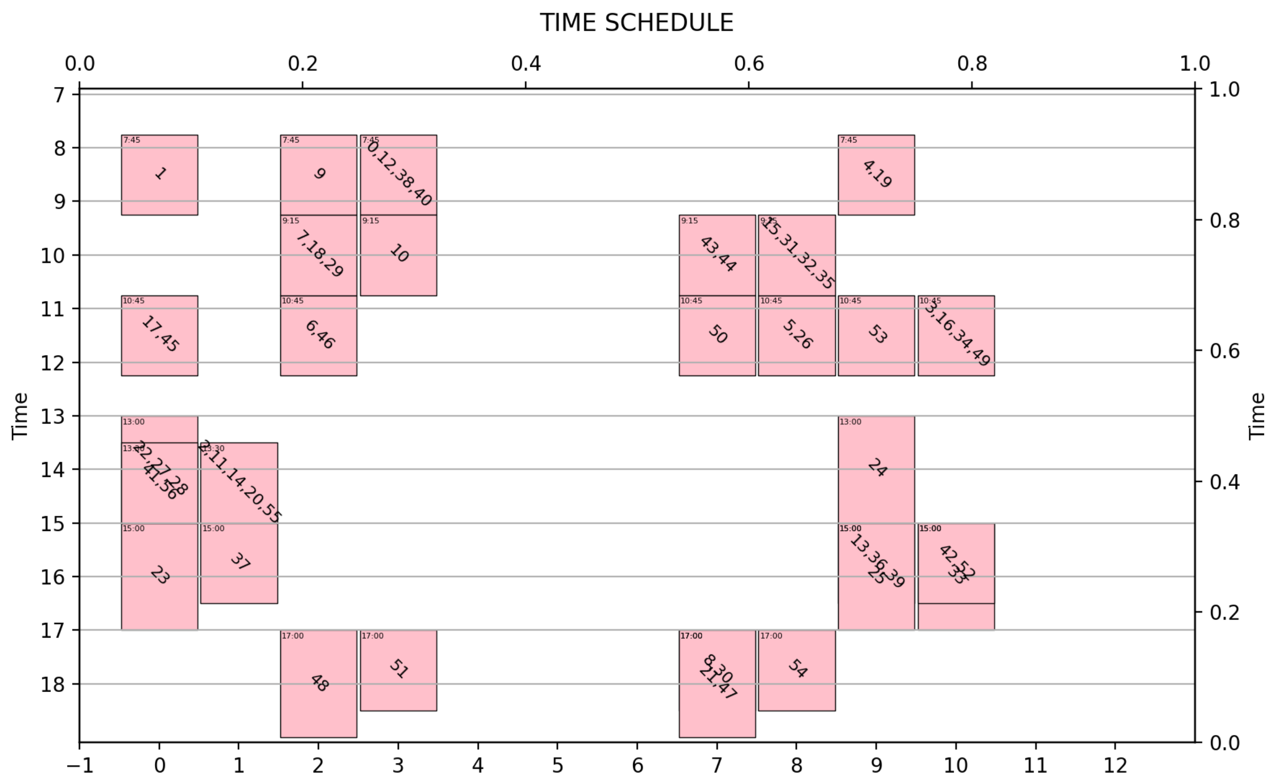
در این فرمولبندی ایده هایی از دو مدل پیشین را ترکیب کردیم تا هم ماکسیمم امتحانات دانشجو بین هفته ها را کم کنیم و هم فاصله بین آن امتحانات در هفته را زیاد کنیم. برای این کار از تابع هدف غیرخطی مدل اول هم استفاده کردیم ولی آن را کمی ساده تر کردیم. لازم به ذکر است در پیاده سازی تابع هدف در فایل project1_v1.6.gms روی اندیس های t و t’ مقداری شرط اضافه وجود دارد. این شرط ها برای کم کردن جمله های در تابع هدف با توجه به شرایط خاص داده ها است. به عبارت دیگر از عدم وجود امتحان ها در پایان هفته در هر دو تابع هدف استفاده شده است تا مدل سریع تر حل شود.

این مدل را برای بازه ۴ هفته ای به مدت ۶ ساعت با بهینه یاب BARON حل کردیم به Integer Solution رسیدیم.

(فایل project1_v1.6.gms) برنامه ریزی انجام شده توسط این مدل به صورت زیر است: (فایل مکمل در پوشه outputs/t4)



نکته قابل توجه این بود که این مدل را بدون ۱۱۸ دانشجو مشکل ساز و برای بازه دو هفته نیز حل کردیم و مدل در کمتر از یک ربع به جواب بهینه زیر رسید: (فایل مکمل در پوشه outputs/t5)



نکاتی در خصوص اجرا کد پایتون:

- تمامی کتابخانه های مورد نیاز در فایل requirements.txt قرار دارند و با دستور *pip install -r requirements.txt* قابل نصب اند.
- برای تسهیل در اجرای کد یک ماژول با نام time travel توسعه داده شده است که استفاده از آن تاثیری در حل مدل ندارد و صرفا برای راحتی بیشتر قبل از اجرای مدل تاریخ سیستم را به عقب می برد و بعد از اجرا تاریخ را به مقدار درست برمی گرداند. در زمان نوشتن این متن، ماژول گفته شده تنها برای سیستم عامل ویندوز کاربرد دارد.
- برای اجرای مدل بدون در نظر گرفتن یک دانشجو خاص کافی است اندیس **j** آن دانشجو در فایل problematic_students.txt در یک خط جدا آورده شود و همچنین در کد پایتون متغیر بولین، exclude_problematic_students برابر True شود.
- برای فرمول بندی دوم و سوم باید کد project1_modified_target.py و برای فرمول بندی اول باید کد project1_single_target.py اجرا شود.

مشخصات دستگاه:

مشخصات سیستمی که جواب ها با آن گزارش شده است به شرح زیر است:

Model	MSI GS65 Stealth 9SF
OS	Microsoft Windows 10 Enterprise
CPU	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2601 Mhz, 6 Core(s), 12 Logical Processor(s)
Memory	DDR4 – 16.0 GB