Group Members: Mei Linlan, Xu Duo, Xinyue Cao, Xuewen Lu, Abhishek Sharma

DSO 570: Interim Deliverable 1 Report

Analysis Direction

The goal of this project is to optimize the current course scheduling system at USC Marshall. In the future, the demand for Marshall courses is expected to rise but the school doesn't have enough land resources to build more classrooms. Thus, we need to figure out a way to efficiently utilize the available capacity of classrooms. Optimal scenarios for stakeholders, including students, faculty, program, department, and management may not be the same. We came up with different metrics to measure "goodness" of each stakeholder. However, based on the data resources and the possibility of achievement, we decide to focus on the department.

There are three reasons why we chose department as our analysis direction. Firstly, we have a good amount of data that tells us the capacity of each classroom and the course schedule in each department. Most data is numerical so we can understand and interpret it easily and make decisions clearly and logically by manipulating the data. In addition, the optimal scenario generated for the department is actionable. For example, it is feasible to adjust the time slots of some courses within one department. We expect that the efficiency of new course scheduling system can be improved a lot and the distribution of time slots can be done more efficiently in a manner most suitable to professors as well as the students alike. The last reason is that the relationship among departments, students and faculty is quite close. We can generate value for departments while satisfying the students and faculty at the same time.

Metric Generation

After cleaning and analyzing the available data, we started with the file "Marshall_Course_Enrollment_1516_1617.xlsx" which has data for each course's classroom location, department, and most importantly, time. Based on the description of the project, the majority of students and faculty don't prefer classes during early morning or late evening hours. Therefore, we assumed the time period from 9:00 am to 5:00 pm as the 'preferable' or 'prime time' during which all the professors and students would prefer the classes to be scheduled.

Since the prime time is limited to 8 hours for each classroom, we have defined the "goodness" as allocating the maximum possible courses in the prime time. It can also be defined as maximizing the utilization of prime time for each department and achieving fairness of time-slots allocation among all departments.

Based on this objective, the important variable we wanted to generate is the amount of course time that doesn't fall into the prime time, for each department. In order to get this variable, we first calculated the time difference of each course that was not scheduled in the prime time. For example, the ACC 201 course from marketing department is from 6:00 pm to 7:50 pm on each Monday and Wednesday. We calculated the total time difference as 1.83 hours ((60+50) minutes

/ 60 minutes = 1.83 hours) since the class is after 5:00 pm (later than prime time) and twice a week.

After generating the time difference for each course, we used Python to sum them up and grouped the results by each department and named the variable as “Total Time Difference”. Then we calculated the total course hours for each department so that we could make suggestions based on department. Additionally, since the total course hours per week for each department differ from one another, we also calculated the percentage of total course hours that fall outside the prime time window with respect to the total hours. The idea is that it makes more sense to compare each department with the ratio of total time difference divided by total course hours.

Both the total course hours and ratio for each department are shown in the below table and we plan to use the ratio to define the “efficiency”.

Analysis of available data:

The below table is the created according to the steps discussed above:

Department	Total Time Difference	Total Hours	Ratio
ACCT	545.00	1898.75	28.70%
FBE	548.83	1389.00	39.51%
BUCO	471.50	1154.33	40.85%
MKT	363.50	796.33	45.65%
DSO	689.67	1219.67	56.55%
MOR	659.50	1149.17	57.39%
BAEP	534.50	792.83	67.42%
IBEAR Program	0.00	24.67	0.00%
EMBA Program	21.17	870.00	2.43%
GEMBA Program	35.50	658.00	5.40%
UG Program	6.00	74.83	8.02%
FT MBA Core	2.00	8.00	25.00%
FT MBA Program	26.00	104.00	25.00%
FIM Program	8.00	29.33	27.27%
OMBA Program	81.50	287.00	28.4%
PM MBA Program	59.00	61.00	96.72%
G Program	25.67	26.83	95.65%

Besides the seven main departments mentioned in the description of project, we find several programs from the data that cannot be assigned to any certain department. So we have included them in the table. In order to guarantee the comprehensiveness of our assumption, we have to pay attention to the programs which may only start classes at a certain time and can hardly be rescheduled due to their characteristics.

Among these programs, G Program and PM MBA Program have extremely high ratios so we look into the detail of these programs. G Program stands for general education program and it has less data about class time in the dataset with most of the classes taking place in office. Based on the schedule we have for G Program, the ratio of class out of prime time is nearly 96%.

Another "unusual" program is PM MBA Program. As we can see from the table, over 96% percent of the course hours are not in the prime period. Since it's a part-time MBA program that mostly opens at night, it may fail to meet the "prime time" assumption. Thus, we decide to regard these two programs as programs that don't have to be scheduled in the prime time. Converse to the assumption of our metric, we'd like to reschedule all the course hours of these two programs outside the prime time window to give more chance to other programs without harming the interest of these two. Additionally, OMBA Program stands for online MBA program and so it does not require any classrooms. Since our focus point of our optimization problem is related to the classroom utilization rate, OMBA Program falls out of scope.

Based on the data analysis, we find that the ratios of non-prime-time course hours are quite different among the departments. For instance, the department BAEP has a ratio of 67.42% while ACCT's ratio is lower than 30%. Among the programs, IBEAR has all the courses scheduled in prime time and the ratio of EMBA and GEMBA are also comparably low. The inefficiency is defined as the inequality of the ratios among departments. As per our understanding, this disparity may adversely impact some departments as some potentially good courses may not have sufficient students due to odd timings. Also, this will impact all other stakeholders. If the time difference ratio of each department is minimized and is comparable to other departments from year to year, we define the schedule to be "efficient". Therefore, we can improvise the current system by assigning more prime time slots to the departments who had comparably high ratio last year. In this way, we can balance the resource allocation among the departments. Also, we hope that we can help Marshall school of business make better use of the classrooms during prime time so that both the students and faculty can benefit.

From the perspective of our optimization problem, the seven major departments are our priority. Now that we have the total time difference and ratio for each department and program, we learn that the average time difference is 544.6 hours and the average ratio is 48.01% for seven major departments last year while the average difference is 14.1 hours and the average ratio is 13.3% for the programs excluding the three unusual ones we mentioned above . Our goal is to prioritize those departments with high ratios such as DSO, MOR and BAEP and to reduce the average ratio of seven departments when making the new schedule.

Thus, in our next step we plan to focus on the optimization metric we have developed with emphasis on the main seven departments' utilization of prime time, followed by other small daily programs and programs that cannot rely on "prime time" assumptions like PM MBA, G Program.

Justification of Appropriateness

Computable: The metric is easily computable. We calculate the ratio of total time difference of each department (called "Ratio"). The exact calculation steps are given above. The metric is free from any ambiguity and despite the fact that the number of courses, the number of students and the number of students per course will change in any department every year, the metric calculation will remain the same.

Actionable: The key objective of the scheduling department is to schedule the courses in a manner that the departments' as well as the students' needs are accommodated. Since the metric

uses the time variance to calculate the underlying 'utility' or 'satisfaction', optimally scheduling the most sought after courses in a more suited or preferable time slot (by minimizing the time difference) can benefit the most number of departments, faculty, and students. This is actionable because the ratio of time difference can be adjusted by changing the schedule of courses in the prime time window.

Simple: The metric is very straightforward. It is very easy to understand and the calculation is very simple. It is also very intuitive in a way that the notion of time is quite obvious.

Enlightening: The reason we picked this metric is that it addresses multiple issues at once. Minimizing the ratio means scheduling maximum courses in the prime time which implies that the professors, students and departments will all be satisfied. In addition, minimization also means that classrooms will be optimally used. So, this metric helps address all the stakeholders in a unified manner.

Appendix

Description of Departments and Programs	
Department	**Description**
ACCT	Accounting
FBE	Finance and Business Economics
BUCO	Business Communication
MKT	Marketing
DSO	Data Sciences and Operations
MOR	Management and Organization
BAEP	Business Entrepreneurship
IBEAR Program	International Business Education and Research MBA
EMBA Program	Executive MBA Program
GEMBA Program	Global Executive MBA Program
UG Program	Undergraduate Program
FT MBA Core	Full-Time MBA
FT MBA Program	General Education Program
FIM Program	Food Industry Management Program
OMBA Program	Online MBA Program
PM MBA Program	Part Time MBA Program
G Program	Undergraduate Program

Appendix Python Code

March 6, 2018

```
In [7]: import pandas as pd
        import numpy as np
```

```
In [37]: schedule=pd.read_excel('Marshall_Course_Enrollment_1516_1617.xlsx')
         cancelled=pd.read_excel('Cancelled_Courses_1516_1617.xlsx')
         master=schedule.append(cancelled)
         master.to_csv('Merged_Enrollment.csv')
         merged = pd.read_csv('Merged_Enrollment.csv')
```

```
In [38]: merged.head()
```

```
Out[38]:    Unnamed: 0    Course Course Prefix Course Suffix Department  \
         0           0  ACCT-370          ACCT           370       ACCT
         1           1  ACCT-370          ACCT           370       ACCT
         2           2  ACCT-370          ACCT           370       ACCT
         3           3  ACCT-370          ACCT           370       ACCT
         4           4  ACCT-371          ACCT           371       ACCT

           First Begin Time First Days First End Time   First Instructor  \
         0         10:00:00          F       11:50:00  Hopkins, Merle, W
         1         08:00:00         MW       09:50:00  Hopkins, Merle, W
         2         10:00:00         MW       11:50:00  Hopkins, Merle, W
         3         12:00:00         MW       13:50:00  Hopkins, Merle, W
         4         10:00:00          F       11:50:00               NaN

            First Instructor UID                ...                   \
         0          3.783354e+09                ...
         1          3.783354e+09                ...
         2          3.783354e+09                ...
         3          3.783354e+09                ...
         4                   NaN                ...

           Second Begin Time Second Days  Second End Time  Second Instructor  \
         0               NaN         NaN              NaN                NaN
         1               NaN         NaN              NaN                NaN
         2               NaN         NaN              NaN                NaN
         3               NaN         NaN              NaN                NaN
         4               NaN         NaN              NaN                NaN
```

```
  Second Instructor UID  Second Room  Section Session   Term  \
0                   NaN          NaN    14029       1  20153
1                   NaN          NaN    14025       1  20153
2                   NaN          NaN    14026       1  20153
3                   NaN          NaN    14027       1  20153
4                   NaN          NaN    14044       1  20153

                                 Title
0  External Financial Reporting Issues
1  External Financial Reporting Issues
2  External Financial Reporting Issues
3  External Financial Reporting Issues
4   Introduction to Accounting Systems

[5 rows x 27 columns]
```

```
In [15]: import pandas as pd
         import numpy as np
         def convert(inputTime):
             # Complete your code here
             try:
                 hh,mm,ss=inputTime.split(':')
                 ans=int(hh)+int(mm)/60+int(ss)/3600
             except:
                 ans=np.nan
             return ans
```

```
In [88]: ans = []
         for index,row in merged.iterrows():
             # Obtain the corresponding column of each row
             course=row['Course']
             dep=row['Department']
             days=row['First Days']
             beg=convert(row['First Begin Time'])   # Convert the begin time strings into deci
             end=convert(row['First End Time']) # Convert the begin time strings into decimal
             times=len(str(days))
             # Skip rows in which beg and end are np.nan (not a number), and in which the room
             if np.isnan(beg) or np.isnan(end):
                 continue     # Command to skip this iteration of the loop
             ans.append([course,dep,days,times,beg,end])
```

```
In [86]: df = pd.DataFrame(ans,columns=['Course','Department','Day','Times','Start','End'])
         df.shape
```

```
Out[86]: (5487, 6)
```

```
In [87]: df['diff'] = (np.where(df['Start']<9, 9-df['Start'], 0)+
                       np.where(df['End']>17, df['End']-17, 0))*df['Times']
```

```
df['tot'] = (df['End']-df['Start'])*df['Times']
department=df.groupby(['Department'])
department.sum()[['diff','tot']]
```

Out[87]:

Department	diff	tot
ACCT	545.000000	1898.750000
BAEP	534.500000	792.833333
BUCO	471.500000	1154.333333
DSO	689.666667	1219.666667
EMBA Program	21.166667	870.000000
FBE	548.833333	1389.000000
FIM Program	8.000000	29.333333
FT MBA Core	2.000000	8.000000
FT MBA Program	26.000000	104.000000
G Programs	25.666667	26.833333
GEMBA Program	35.500000	658.000000
IBEAR Program	0.000000	24.666667
MKT	363.500000	796.333333
MOR	659.500000	1149.166667
OMBA Program	81.500000	287.000000
PM MBA Program	59.000000	61.000000
UG Programs	6.000000	74.833333

In [81]:
```
department.Course.nunique()
```

Out[81]:
```
Department
ACCT              64
BAEP              38
BUCO              23
DSO               63
EMBA Program      15
FBE               72
FIM Program        4
FT MBA Core        1
FT MBA Program     3
G Programs         3
GEMBA Program     10
IBEAR Program      2
MKT               40
MOR               53
OMBA Program       7
PM MBA Program     2
UG Programs        8
Name: Course, dtype: int64
```