

## Using An Engineering Analysis Tool for Department Administration

**Dr. Hugh Jack P.E., Western Carolina University**

Dr. Jack is the Cass Ballenger Distinguished Professor of Engineering and Department Head of the School of Engineering and Technology within Western Carolina University. His interests include robotics, automation, and product design.

# Using An Engineering Analysis Tool for Department Administration

## Abstract

The paper describes a basic application created using Matlab to assist in academic scheduling of technical programs. The work presented is the first iteration of a larger system. For this first phase of the project, the system uses standard (draft) schedule report spreadsheets. These are processed to calculate individual faculty loads, detect conflicts, produce visual schedules, track unstaffed courses, banked credit, etc. More importantly, the analysis uses information about the structure of the curriculum to support cohort oriented scheduling.

The project has been implemented in Matlab using Tables. Although other platforms might be better suited to this type of data analysis, the use of Matlab makes it easier to modify and extend in Engineering schools. This is in major part due to the widespread use of Matlab in Undergraduate and Graduate curriculum. The programs described in this paper are available for other schools that would like to use and/or modify for their own purposes.[1]

## Introduction

Administrative tasks like scheduling require substantial work doing iterations of menial tabulations, analysis, and revision. Most colleges and universities have software that helps with these tasks, but it is not well suited to a technical curriculum with extensive prerequisites, laboratory/equipment/software constraints, and faculty availability. The result is a lot of manual processing of spreadsheets and timetables.

At Western Carolina University (WCU) there are five residential undergraduate programs that share a number of courses. In addition, there are two undergraduate programs that are offered off-campus and a residential and an off-campus Master's program. Growth over a few years has outpaced the acquisition of new faculty, space, and equipment. The conflict of accommodating more students within the constraints of the limited resources is making scheduling increasingly more difficult. This work was inspired by standard scheduling duties such as finding overlapping course times, faculty load calculation, keeping track of adjuncts, dealing with schedules to accommodate students out of sequence, and integration with math and physics courses.

Tools were created to perform data manipulation and analysis for standard scheduling tasks. These included calculations often foreign to other non-STEM departments such as differing credit for teaching laboratories. At the point of writing the tools would automatically generate schedules by individual program and indicate possible conflicts. Note: it is not currently the authors' goal to automate schedule creation, although this might occur later.

The toolchain used for the work was based on Matlab. Matlab provides excellent capabilities for data analysis. (If needed, free alternatives exist including Octave and Scilab.) More importantly, Matlab programming is widely used in technical schools so the expertise to modify, use, and extend can be easily found. Scripts were written to read data from multiple spreadsheets in Comma Separated Variable (CSV) formatted files. The files contained the routine output of a draft schedule for any given semester. The scripts manipulate the data and update or generate new spreadsheets. The scripts were written using standard Matlab functions and tables. It is worth noting that previous work uses custom written software for Windows [2] and a web based software tool [3]. Both of these are very capable, and in many ways better, than the current tool. However, both require a greater knowledge of computer programming to adapt.

As an aside, the obvious question is ‘Why not use the existing data system?’ At WCU the student records system is based around Banner, by Elucian. Like most campus data management systems, it is built around a database with numerous tables. In an ideal world, there would be very good university-level resources for developing analysis scripts or at least access to the database directly through Sequential Query Language (SQL) requests. But, at this point, the system cannot process the multiple factors needed to analyze a schedule without spreadsheets.

The methodology is to download required data from Banner Reports in CSV formatted spreadsheets. These reports include course lists for given semesters, grades for all students by semester, etc. The input and output data files are stored in CSV spreadsheets. A directory structure is used for organization and simplicity. This means that data values can be easily read with any spreadsheet program.

The paper describes the tool in enough detail that it can be adapted for use in other engineering and technology schools with little to no program modification.

### **Schedule Information**

At most universities, the schedule is available in a spreadsheet format. A sample of the spreadsheet used for this work is shown in Figure 1. As expected the column names are somewhat cryptic, but over time they begin to make sense. The spreadsheet shown is for a past semester, after registration, with enrolled students shown. This report (spreadsheet) is easily generated for individual programs or the entire university.



In the application, these columns are renamed to standard titles and new columns are added. The program has been written in a modular format to allow easy substitution of titles. The output spreadsheet is shown in Figure 2. In addition to renaming the columns, a few basic calculations have been made. This includes changing the time from AM/PM to military formats. The time durations have been calculated. This requires some correction for academic practices such as 50 minutes = 1 hour and a 3-hour laboratory might be 2 hours and 45 minutes. It is worth noting that the durations of ECET 452 and EE 311 are not an even or half and the times for these courses will need to be adjusted. The weight reflects that laboratory time is discounted to 50% for laboratories, a common practice. Once students are enrolled the SCH (Student Credit Hour) and FTE (Full Time Equivalent) will be calculated.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	
1	CRN	subject	number	section	campus	meeting	credits	link	title	limit	enrolled	available	format	days	t_start	t_end	room	seats	instructor	duration	meeting	weight	SCH	FTE	FCR	banked	year	notes	comments
2	80042	ECET	221	75	M	F	1		Digital Integrated Circuit Lab	30 (0)	0	30	LAB	W	15:35	17:15	TBA	NaN	Yanik P	2	1	0.5	0	0	1	0	0		
3	80043	ECET	231	1	M	F	4	G3	Circuit Analysis I	26 (0)	0	26	LEC	M	9:05	10:45	CAT 219	26	Ritenour A	2	1	1	0	0	2	0	0		
4	80044	ECET	231	25	M	F	0	G5	Circuit Analysis I-LAB	26 (0)	0	26	RCT	F	9:05	10:45	CAT 219	26	Ritenour A	2	1	0.5	0	0	1	0	0		
5	80048	ECET	231	30	M	F	0	G5	Circuit Analysis I-LAB	26 (0)	0	26	LAB	W	9:05	10:45	CAT 221	32	Ritenour A	2	1	0.5	0	0	1	0	0		
6	80049	ECET	301	1	M	F	3	F6	Electrical Systems	36 (0)	0	36	LEC	T	12:35	14:15	CAT 210	36	Denton J	2	1	1	0	0	2	0	0		
7	80051	ECET	301	30	M	F	0	F7	Electrical Systems-LAB	16 (0)	0	16	LAB	R	12:20	14:00	BL 366	24	Denton J	2	1	0.5	0	0	1	0	0		
8	80054	ECET	301	31	M	F	0	F7	Electrical Systems-LAB	16 (0)	0	16	LAB	F	14:30	16:10	BL 366	24	Denton J	2	1	0.5	0	0	1	0	0		
9	80058	ECET	301	32	M	F	0	F7	Electrical Systems-LAB	16 (0)	0	16	LAB	F	12:20	14:00	BL 366	24	Denton J	2	1	0.5	0	0	1	0	0		
10	80058	ECET	321	25	M	F	0	FJ	Circuit Analysis II-RCT	23 (0)	0	23	RCT	W	18:00	19:40	BL 355	28	Yan Y	2	1	0.5	0	0	1	0	0		
11	80059	ECET	321	30	M	F	0	FK	Circuit Analysis II-LAB	23 (0)	0	23	LAB	F	13:25	15:05	BL 355	28	Yan Y	2	1	0.5	0	0	1	0	0		
12	80057	ECET	321	70	M	F	4	FI	Circuit Analysis II	23 (0)	0	23	LEC	T	15:35	17:15	BL 355	28	Yan Y	2	1	1	0	0	2	0	0		
13	80062	ECET	431	1	M	F	4	FT	Microcontroller Interfacing	23 (0)	0	23	LEC	TR	12:35	14:15	TBA	NaN	Tay P	2	2	1	0	0	4	0	0		
14	80067	ECET	431	30	M	F	0	FJ	Microcontroller Interface-LAB	23 (0)	0	23	LAB	M	12:20	14:00	TBA	NaN	Tay P	2	1	0.5	0	0	1	0	0		
15	80071	ECET	446	1	M	F	4	G1	Modern Power Systems Analysis	23 (0)	0	23	LEC	TR	12:35	13:50	BL 370	23	Karayaka H	1.5	2	1	0	0	3	0	0		
16	80073	ECET	446	75	M	F	0	G2	Modern Pwr Sysms Analysis-LAB	23 (0)	0	23	LAB	T	17:30	19:10	BL 366	24	Karayaka H	2	1	0.5	0	0	1	0	0		
17	80077	ECET	452	1	M	F	4	FV	Control Systems	23 (0)	0	23	LEC	MW	12:20	13:35	BL 370	23	Tay P	1.5	2	1	0	0	3	0	0		
18	80081	ECET	452	30	M	F	0	FW	Control Systems-LAB	23 (0)	0	23	LAB	W	14:00	15:30	TBA	NaN	Tay P	1.8	1	0.5	0	0	0	0.9	0		
19	80084	EE	200	1	M	F	3		Computer Utilization	40 (0)	0	40	LEC	MW	13:25	14:40	TBA	NaN	STAFF	1.5	2	1	0	0	3	0	0		
20	80089	EE	200	2	M	F	3		Computer Utilization	30 (0)	0	30	LEC	MW	9:45	11:00	TBA	NaN	STAFF	1.5	2	1	0	0	3	0	0		
21	80090	EE	200	80	ABP	F	3		Computer Utilization	0 (0)	0	0	LEC	TBA	TBA	TBA	NaN	STAFF	3	1	1	0	0	0	0	0			
22	80091	EE	201	1	M	F	3		Network Theory I	35 (0)	0	35	LEC	MWF	8:00	8:50	TBA	NaN	Huang Y	1	3	1	0	0	3	0	0		
23	80092	EE	211	30	M	F	1		Instrumentation & Netwks Lab	32 (0)	0	32	LAB	F	14:30	17:00	CAT 221	32	Adams R	3	1	0.5	0	0	1.5	0	0		
24	80094	EE	221	2	M	F	3		Logic Systems Design I	27 (0)	0	27	LEC	MWF	10:10	11:00	TBA	NaN	Yanik P	1	3	1	0	0	3	0	0		
25	80093	EE	221	70	M	F	3		Logic Systems Design I	40 (0)	0	40	LEC	MW	18:00	19:15	TBA	NaN	Yanik P	1.5	2	1	0	0	3	0	0		
26	80095	EE	311	75	M	F	1		Systm & Electronics Lab	28 (0)	0	28	LAB	M	17:30	19:30	CAT 221	32	Adams R	2.4	1	0.5	0	0	1.2	0	0		
27	80096	EE	321	1	M	F	3		Electromagnetic Fields	28 (0)	0	28	LEC	MWF	9:05	9:55	TBA	NaN	Huang Y	1	3	1	0	0	3	0	0		
28	80097	EE	331	1	M	F	3		Fundamentals Ectnes & Semicon	28 (0)	0	28	LEC	TR	8:00	9:15	TBA	NaN	Yang W	1.5	2	1	0	0	3	0	0		
29	80098	EE	351	1	M	F	3		Systems Analysis I	28 (0)	0	28	LEC	TR	14:30	15:45	TBA	NaN	Huang Y	1.5	2	1	0	0	3	0	0		
30	80099	EE	411	1	M	F	3		Analog Communication System	30 (0)	0	30	LEC	MWF	12:20	13:10	CAT 210	36	Huang Y	1	3	1	0	0	3	0	0		
31	80100	EE	421	1	M	F	3	FN	Digital Sys Design	28 (0)	0	28	LEC	R	12:35	14:15	CAT 219	26	Yanik P	2	1	1	0	0	2	0	0		

Figure 2 - The processed schedule spreadsheet

## Scheduling

As we know scheduling can be complicated in simple environments. In technical programs, these problems are exacerbated by factors including,

- Long prerequisite strings where missing one course can delay graduation by a year.
- Limited offerings of critical classes including once per semester.
- Corequisite course requirements.
- Constraints like shared laboratories, equipment, and software.
- Multiple programs with shared courses.
- Courses for student cohorts must not overlap. Ideally accommodating students out of sequence.
- Competition for space with other programs.
- Demand for blocked times including department meetings.
- Coordination with other departments including Physics and Math.
- Distances between spaces and transit time.
- Faculty preferences and time requirements.

The common, although not universal, approach to scheduling is - begin with a schedule from a previous semester. The schedule is then adjusted to accommodate section addition/deletion/modification, room changes, capacity changes, instructors, etc. The schedule is then reviewed to find and correct conflicts. While a spreadsheet is adequate, a printed schedule with all courses shown for a program makes it very easy to identify problems. Likewise, a schedule printed for each instructor will identify conflicts.

The schedule is generated using the updated spreadsheet. The result is saved as an SVG (Scalable Vector Graphics) file, shown in Figure 3. In this example, there is an overlap on Monday and Wednesday mornings. But, the overlapping course is a prerequisite for the other courses and will not be a problem. For complex schedules, these can become quite busy but the SVG format allows zooming without image quality degradation.

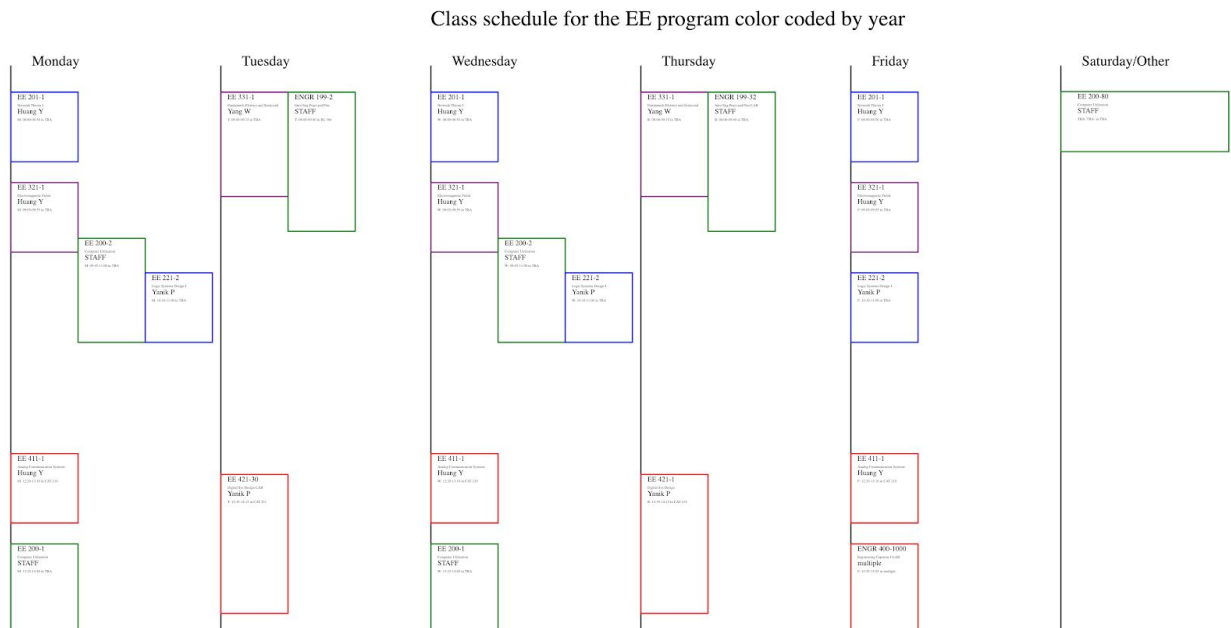


Figure 3 - Sample course schedule

Faculty schedules are relatively easy to generate using the same method. Figure 4 shows a schedule for one faculty member. In their case, there is a time conflict on Monday.

## O\_Schedules/Schedule\_Tay P\_Fall\_2019.svg

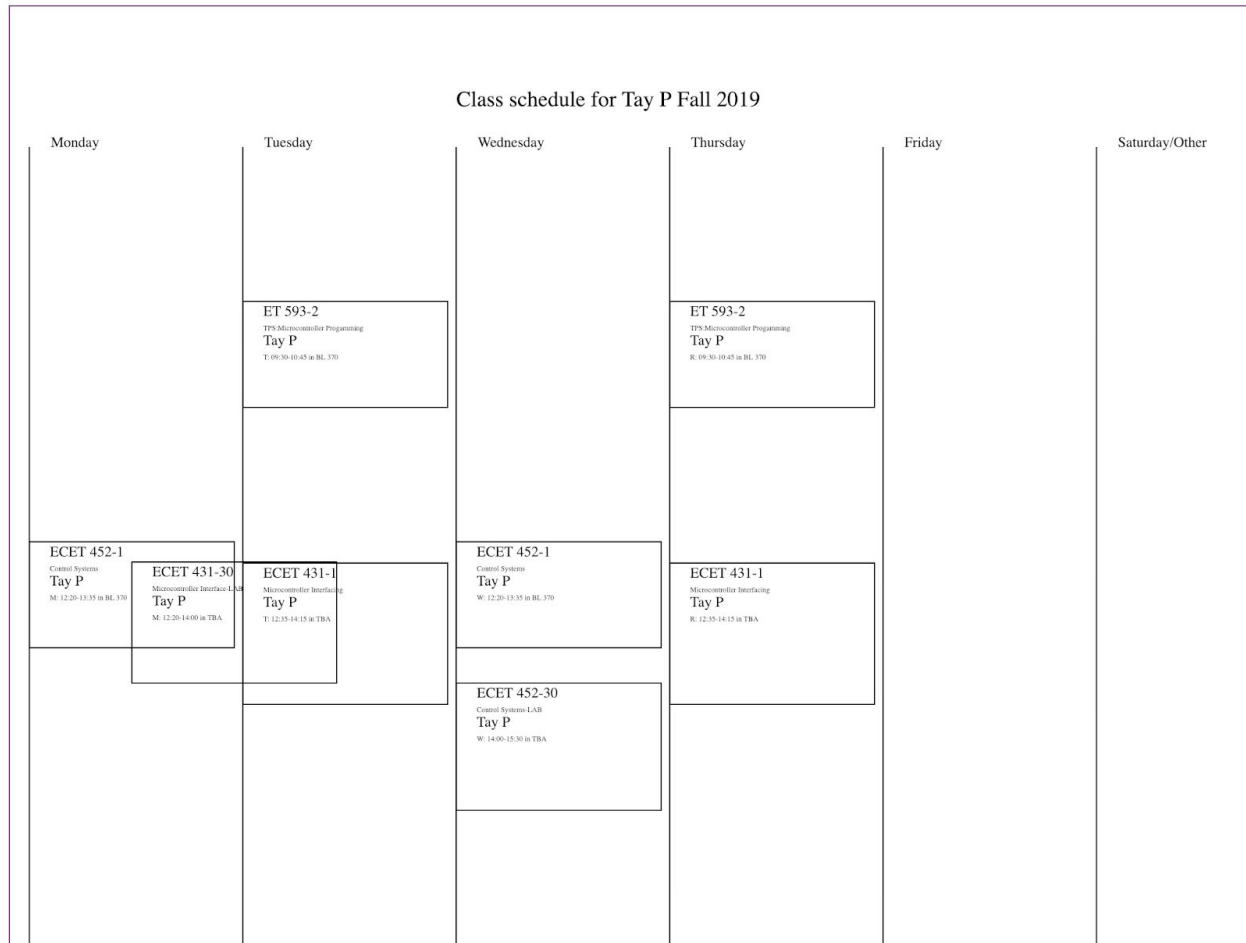


Figure 4 - Faculty schedule output

The process of schedule generation takes a few seconds. Changes to the schedule likewise take a few minutes for simpler tasks. This allows a few full iterations of the schedule within an hour.

The process workflow is,

1. Download the 'Banner Report' spreadsheets with the data. Examples of these reports include,
2. Files are renamed to reflect Semester and Year and moved to various directories for analysis. Subdirectories are used to aid organization and reduce clutter.
3. Scripts are run and the results are stored as spreadsheets (CSV), images (SVG), and 'documents' (HTML).
4. (Scheduling) the schedules for the programs and faculty are examined. If needed, changes are made. The results are then rerun (back to step 1.)
5. The numbers for session lengths, missing instructors, and similar problems are identified and corrected if possible.

Although a humble start, these tools form the basis for detecting constraints automatically and provide a framework for identifying faculty preferences such as full days for research. These will be used in a future iteration of the program to support automatic iterative schedule adjustment. This type of work has been done with many methods included genetic algorithms.

## **System Architecture**

Ideally, the software and analysis needs would be handled by the IT and analysis offices on campus. Practically there are limited resources and interest in developing tools for programs that are only a small percentage of the university. Luckily many engineers are comfortable with software and programming.

There are many excellent approaches to implementing this type of system. These include SQL database queries, Python scripts, traditional programming, data analytics software, etc. The method employed for this work was,

1. The data sources are various Comma Separated Variable (CSV) spreadsheets downloaded from the university database (Banner.)
2. Programs were written in Matlab to take advantage of the broad knowledge within engineering and technology schools.
3. The programs are run using Matlab as '.m' files.
4. The output files are examined as spreadsheets or graphically using a browser to open an HTML file with SVG images.

Of special note is the use of tables in Matlab. Unlike matrices, these allow mixed data types like doubles and strings. It is then easy to address data elements by name instead of column number.

## **Coming Next**

This platform promises to be an excellent vehicle for further analysis for schedule and student data. The use of Matlab also increases the ability for technical schools to modify and maintain the programs. Other features that are currently implemented or in development are listed.

- Automating degree audits to generate suggested schedules for future semesters. Currently, the students can only retrieve a degree audit that shows remaining courses - without prerequisite sequences and recommended semesters.
- Automatic screening for student major admission and progress tracking.
- Detailed tracking of students as they enter and leave the programs.
- (long-term) Automated schedule adjusting.



## References

- [1] Jack, H., *Matlab Software Tools For Schedule Analysis*, Google Drive Directory - <https://drive.google.com/open?id=11k8uBPOe85TiXf4BYAt7WN27ecvbLb4N>
- [2] Mihali, R., & Vamoser, D., & Sobh, T. (2001, June), *Sked: A Course Scheduling And Advising Software* Paper presented at 2001 Annual Conference, Albuquerque, New Mexico. <https://peer.asee.org/9781>
- [3] Danda, P., *A Web-Based System for Course Instruction and Student Advising*, Master's Thesis, Department of Industrial and Management Systems Engineering, Morgantown, West Virginia, 2009