Retrospective of progress made from Aug-May 2025

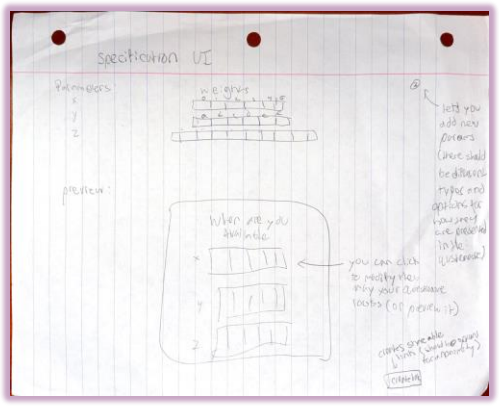
Dakota Lawson

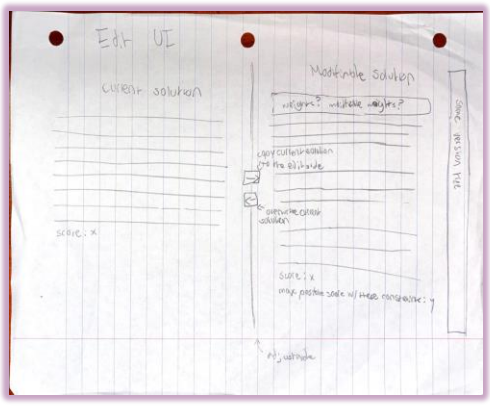
Under the supervision of Dr. Michael Stewart

Throughout the course of this semester, I have worked on developing a project by the name of Expert User Interfaces for Domain Specialist Access to Algorithmic Solutions. This project aims to develop software that enables users to collect data from two parties and apply constraints to draw conclusions—such as determining the optimal schedule for multiple one-on-one meetings or assigning employees to shifts based on staffing requirements and student availability.

Initial Ideas and observations

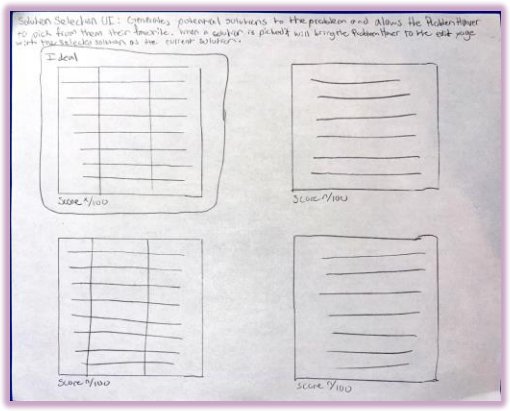
We brainstormed how a web application could provide access to algorithms that solve user's problems intuitively. Initially, we took inspiration from Video editing software. Video editing software tends to separate user interfaces into two parts. A section that allows the user to cut and pick videos and place them into a timeline; followed by a section that allows the user to edit the clips in the timeline. Similarly, we wanted to be able to import data and generate an initial result followed by a more in-depth editing process. As a result of our brainstorming, we ended up with these sketches.

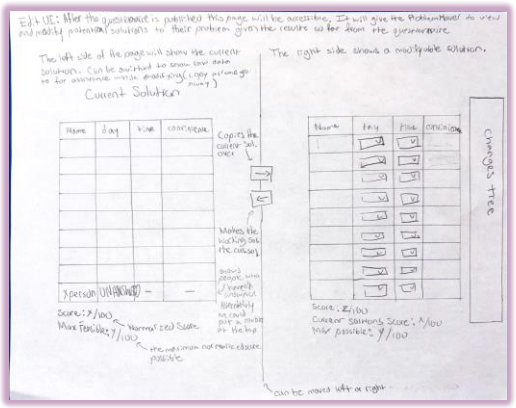




Our initial concept required the user to set weights accociated with the user provided data. Those weights would then be input into the algorithmic engine along with the data. However, we realized that without intimate knowlage of the algorithmic engine, introducing a large amount of direct control without any instruction may be chanllenging to use. Idealy, we want to enable the user to inform the algorithmic engine what soultion they are looking

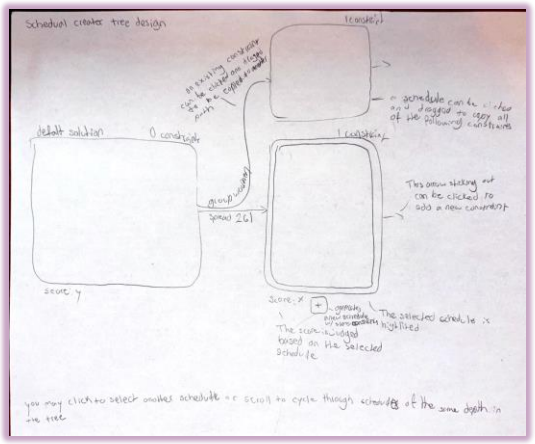
for. To accomplish this, we took inspiration from eye exams. After selecting the weights, the user is presented with a few options to choose from which help the software identify the user's goal. Resulting in the following updated sketches.





We found the eye exam to be imperfect because it leaves a gap in the knowledge of the user. With this system the user will not be able to gain any intuition about how the weights affect the resulting solution. Without understanding the user will likely not be able to generate the solution they are looking for. Instead of focusing on the algorithm understanding the user, we need the user to understand the algorithm.

To understand a complex system, users need to focus on one piece at a time. So, we decided to pivot, giving the user the ability to make one decision at a time. The user would be able to add a constraint and see how it effects the solution to verify their understanding. They then could modify the constraint or continue to add another constraint. We want the user to be able to discover how they can reach the solution one step at a time. A tree-based structure for the user interface would lend itself well to this idea.



Implementation

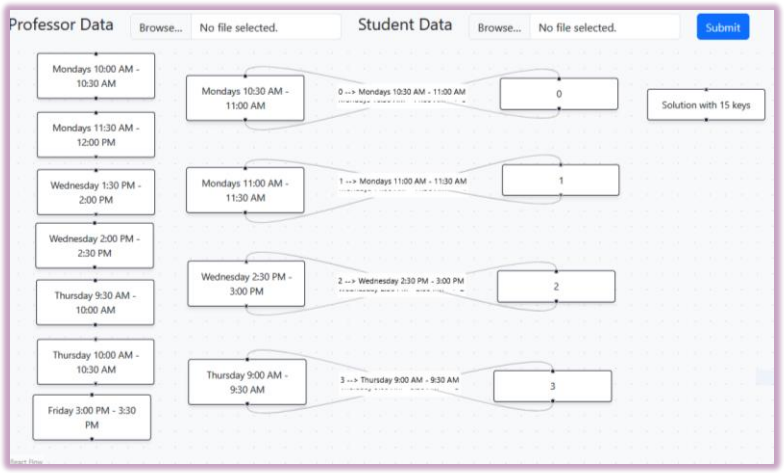
Our goal is to start with something very specific to one application and slowly make it more accessible, allowing the design to shape over time. We began by scheduling availability for 1 on 1 meetings between students and a professor. This is a maximum bipartite matching problem which can be solved in multiple ways. However, simulated annealing has the most potential to be applied to other similar problems that can't be solved as conveniently. We made a web application that allows the user to input two csv files, one detailing a professor's availability and the other the students' availability. The software then finds a solution using simulated annealing and displays the solution as a graph; where nodes are students and time slots, and connections are assignments.

For Example:

```
"What times are you available to meet with Dr. Stewart? (please select all times that you are available)"
Mondays 10:00 AM - 10:30 AM
Mondays 10:30 AM - 11:00 AM
Mondays 11:00 AM - 11:30 AM
Mondays 11:30 AM - 12:00  PM
Wednesday 1:30 PM - 2:00 PM
Wednesday 2:00 PM - 2:30 PM
Wednesday 2:30 PM - 3:00 PM
Thursday 9:00 AM - 9:30 AM
Thursday 9:30 AM - 10:00 AM
Thursday 10:00 AM - 10:30 AM
Friday 3:00 PM - 3:30 PM
```

```
ID,Start time,Completion time,Email,Name,Last modified time,Phone number,What times are you available to meet with Dr. Stewart? (pl
0,02/11/25 14:19:43,02/11/25 14:21:43,dana67@example.net,Mr. Jimmy Sullivan,,740.409.6815x4823,Mondays 10:30 AM - 11:00 AM;Mondays
1,02/07/25 12:34:46,02/07/25 12:43:46,christopherbarnett@example.org,Rhonda Garrison,,001-300-794-2157x19468,Mondays 11:00 AM - 11:
2,02/09/25 12:35:37,02/09/25 12:36:37,scowan@example.com,Cory Jacobs,,+1-328-712-5894x4409,Thursday 10:00 AM - 10:30 AM;Mondays 10:
3,02/10/25 22:33:59,02/10/25 22:35:59,jenniferpittman@example.org,John Smith,,2622333266,Mondays 10:30 AM - 11:00 AM;Thursday 9:00
```

The above inputs produce this output:



Observations

Displaying the information in an uncluttered manner is going to be challenging. Due to the nondeterministic nature of simulated annealing, you may end up with different solutions across multiple runs. However, the tree structure is not well suited to displaying such information in an easily navigable way. Furthermore, the implementation we have created is extremely limited, right now to solve different problems the user would need to modify the source code of the web app to solve their problem.

Future Work

Future work on this project should begin with a literature review. To further explore the space, we could imagine proceeding in a depth-first or a breadth-first manner. If we were proceeding in a depth first manner, the next step would be: iterating on the UX design, implementing more algorithms, and making the elements of the algorithms modifiable in the web app. Alternatively, if we were to proceed breadth-first we could next add a new kind of problem with an associated algorithm or heuristic approach. It could be beneficial to ask students who have applicable problems to modify the web app to satisfy their needs. Then identify commonalities between their solutions and use that to further inform the UX design. Further useful work would be adding manual editing of solutions, implementing a way to export the solution, and a feature that allows the user to lock components of a “solution candidate” before running the algorithm again.

Takeaways

This was my first real experience with developing a web application, so I learned about JavaScript and web development. Web applications rely heavily on asynchronous processing to keep the browser responsive while code runs in the background. JavaScript was developed with this in mind and places a high value on backwards compatibility causing it to have many unique intricacies.

I learned about human computer interaction and the methods by which research is done in this field. Avoid investing a large amount of time in one thing, try many things before deciding. The act of trying those things is more important to research than having a working product. The wizard of oz method is a good example of this, the research is not about how to program a web app, it is about how the people interact with the web app.

This is an interdisciplinary project, and I was able to learn about what aspects of computer science excite me the most. I enjoy thinking about how people interact with software. Further, I like the idea of using math and computer science to help people. I want to make it accessible to both people who like math and those who do not.