COG366 Project Task 9 - Discussion

Our implemented computational model is successful in prioritizing tasks based on three factors: types of tasks, due dates, and difficulties of tasks. A user is able to add new tasks to the belief set and modify the details of existing tasks. When any action that may change the belief set is performed, the model will reconsider the order of the tasks. So the main goal of this model is accomplished. Using this model as a framework, one can easily make modifications to make it fit into a different scenario, e.g., a business determining which orders to fulfill first. Similar to our world, the order that was placed the earliest should be processed first. There are also cases where a customer would pay extra to have express shipping or such, then that order should be moved up to the top of the list. This can be taken care of by specifying the type of the order.

There are a few limitations in our implemented computational model. To start off, we simplified our prioritization factors down to just three as mentioned above. We did not consider the specific amount of time needed for a task, which is actually important in a real world scenario. However, one can claim that time is considered one of the factors, which is the difficulty of a task. Since we did not define difficulties explicitly, when a user is deciding the difficulty of a particular task, one can take the amount of time needed into consideration. Another limitation is that this model does not show any bias, which is something present in humans. When we are deciding which task to prioritize, we always have certain tasks in our heads that we definitely want to get done first, and some that we just want to postpone forever. We do not really have an effective way to show that in the model reasonably. Moreover, this model only depicts the mindset of one person, but everybody may have a unique way to decide one's priority. We do not know if the mindset we have chosen is efficient enough, and we don't really have other systems to compare our model to. Therefore, this model may not be as good when it comes to generality. If we have more time to work on this model, one thing we would definitely like to improve is error handling. In the current version, if a user enters some inputs that

cannot be parsed, the program will end with a "false" without any explanation of failure. To fix this, we will have to add an extra case in the code every time we are reading an input from the user. If the put cannot be parsed, we have to let the user know the correct format of input, and repeat the action of reading input until the program receives a reasonable input and can parse it without any problem.