Graham Lovell

02/02/2025

MSDS 460, Decision Analytics

Assignment 2 - Network Models (Project Management)

Assignment 2 was an individual assignment where we were introduced to network problems, such as shortest path problems, and we were able to utilize linear programming to solve for the shortest path. After being introduced to linear programming in a previous module this allowed us to continue using that knowledge in a new application. For this assignment, I was tasked with determining the best-case, expected, and worst-case estimates for the number of hours needed for a development project that requires 16 different individual tasks. We were given an excel file that listed these 16 tasks and after I filled in time estimates for each task and its best-case, expected, and worst-case scenario I was able to code the information into decision variables, constraints, and objective function within python using the Pulp model to solve for the shortest path. The goal of this assignment was to use linear programming to solve a shortest path problem thus finding the minimum total time solution for this hypothetical development task. This also answers the minimum total cost solution as we simplified the problem to state that the hourly rate was constant across the 16 different development tasks.

To complete this assignment, I chose Python as the language I would use and Jupyter Notebooks as the environment. I chose Jupyter Notebook because the cell-based approach makes debugging easy and once I was happy with my code I just condensed it all into one cell and saved as a .py file. Additionally, I chose Pulp as the model that I used to complete this linear programming task. I chose Pulp because I have previous experience with the library and it is fairly straight forward. Once I had filled in time estimates for each task and its best-case, expected, and worst-case scenario I was able to code the information into decision variables, constraints, and objective function within python using the Pulp model to solve for the shortest path. I created three sets of activity dictionaries, but rather than run the same code three times on a different dictionary I opted to just write out three different minimization problems for each dictionary. Alternatively I could've just looped through each dictionary I created, and as this would be less repetitive I would make this change in a future iteration of this assignment. Once I had the solution for the minimum total time solution in each scenario I multiplied that number of hours by $31.51 which I found to be the average hourly rate for software developers from Payscale.com. The product of those numbers was the minimum total cost solution for each scenario.

My code first prints the solution for the best-case scenario, followed by the expected case, and then the worst-case scenario. The minimum time and cost solutions

for the associated scenarios were found to be 108 hours at $3403.08, 210 hours at $6617.10, and 314 hours at $9894.14. The differences in tasks that could be worked on in parallel changed based on the necessary number of hours in each scenario and can be seen in greater detail within my codes output text file.