Assignment 2: Network Models---Project Management
Ryan Fallon

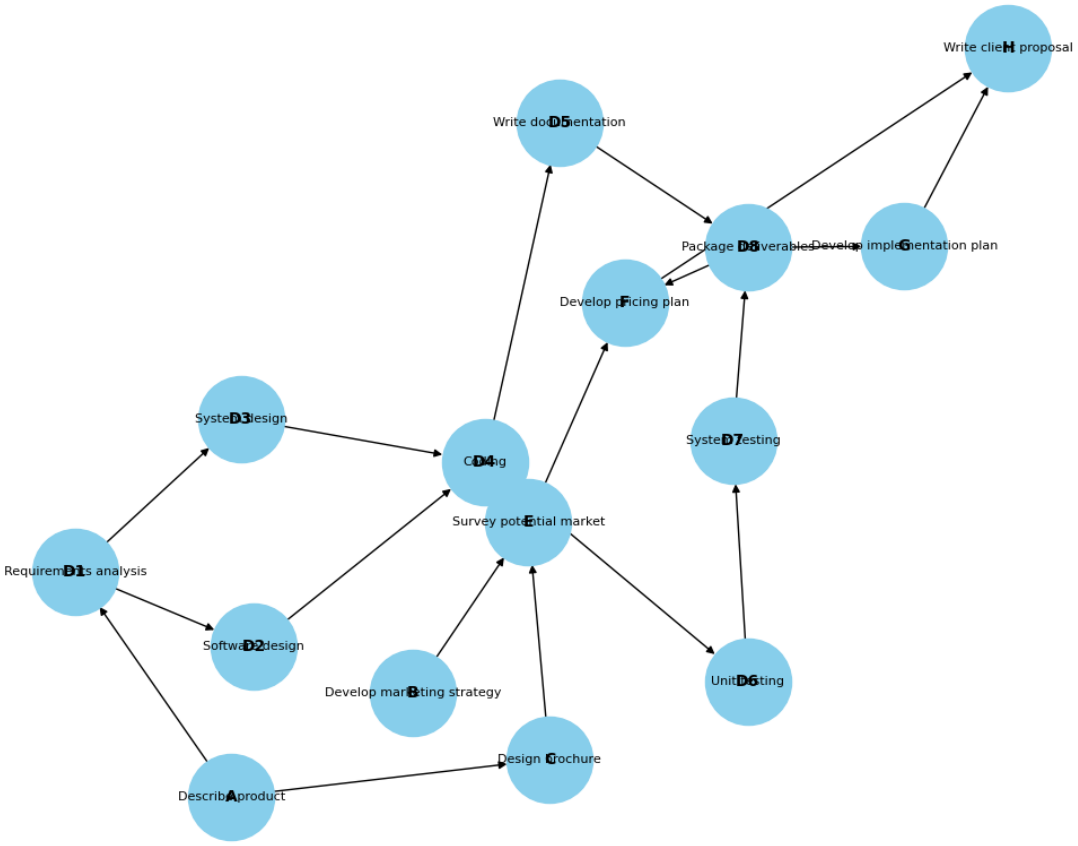
Part 1: Problem Setup

In this project, I developed a preliminary plan to estimate the timeline and resources needed for a restaurant recommendation system. Starting with a list of tasks and dependencies, I assigned conservative estimates for best-case, expected, and worst-case hours, in 0.5-hour increments, to give a realistic range of completion times. Additionally, I allocated team roles, including Project Manager, Frontend and Backend Developer, Data Scientist, and Data Engineer, to each task based on skill requirements.

Some areas of uncertainty remain, particularly regarding the time needed for tasks that depend on data availability, such as requirements analysis and market surveying. Cost estimates could also vary, depending on final contractor rates and any unforeseen complexities. I created a directed graph to visualize task dependencies, identifying the critical path to highlight tasks that cannot be delayed without impacting the project timeline. Tasks like marketing strategy and brochure design can proceed in parallel, providing some flexibility.

Both the updated Excel Spreadsheet and Directed Graph pictured below can also be found in the repository:

taskID	task	predecessor	bestCaseHo	expectedHou	worstCaseH	projectMana	frontendDev	backendDev	dataScientis	dataEngineer
A	Describe product		2	3	4	1				
B	Develop marketing strateg		1.5	2.5	3.5	1				
C	Design broch	A	1	2	3		1			
D	Develop product prototype		5	6	7			1		
D1	Requirement	A	2	3	4	1			1	
D2	Software des	D1	3	4.5	6			1		
D3	System desig	D1	3.5	5	6.5			1		1
D4	Coding	D2, D3	4	5.5	7			1		
D5	Write docum	D4	1.5	2.5	3.5	1			1	
D6	Unit testing	D4	2	3	4			1		
D7	System testir	D6	2	3	4			1		
D8	Package deli	D5, D7	1	2	3	1				
E	Survey poten	B, C	1.5	2.5	3.5	1				
F	Develop pric	D8, E	2	3	4					1
G	Develop impl	A, D8	3	4	5	1			1	
H	Write client p	F, G	2.5	3.5	4.5	1				



Part 2: Model Specification

For the project plan, I set up a linear programming (LP) model using Pulp to minimize the total project duration. The main goal is to determine the shortest time required to complete all tasks, taking dependencies and task durations into account. Since each role is billed at the same hourly rate, minimizing time also minimizes cost, which simplifies the model.

The LP model includes variables for each task's start and end times, with constraints ensuring each task begins only after its predecessors finish. The objective function minimizes the project's total completion time by finding the earliest possible end time for the last task on the critical path. This approach allows us to identify the optimal timeline and manage the resources efficiently.

By applying this LP model, we get a clear path to completing the project in the least amount of time while keeping costs low, assuming consistent hourly rates across team members.

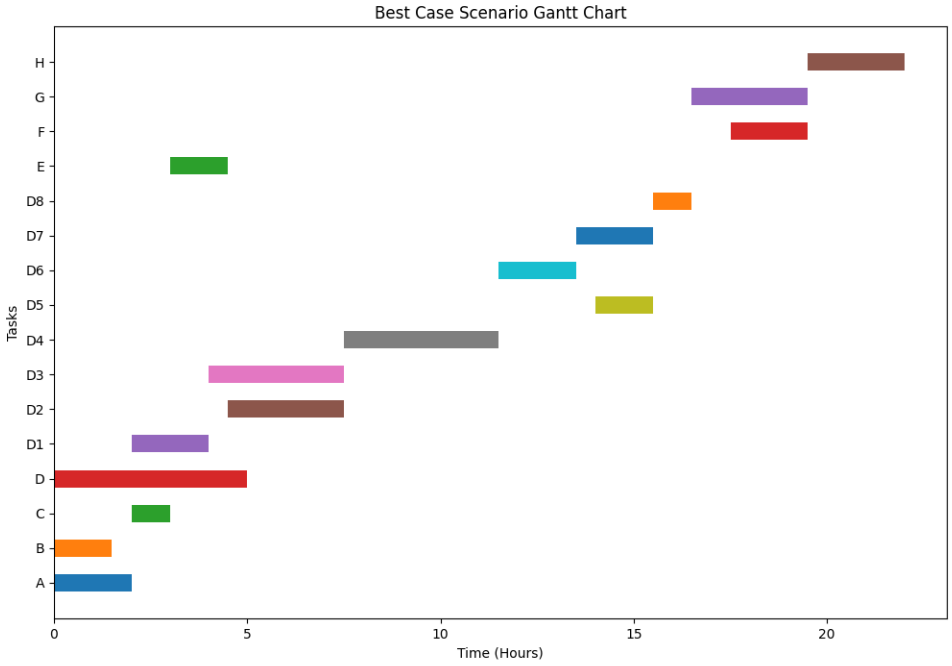
Part 3: Programming

https://github.com/ryano0oceros/projectrefuge-networkmodels460

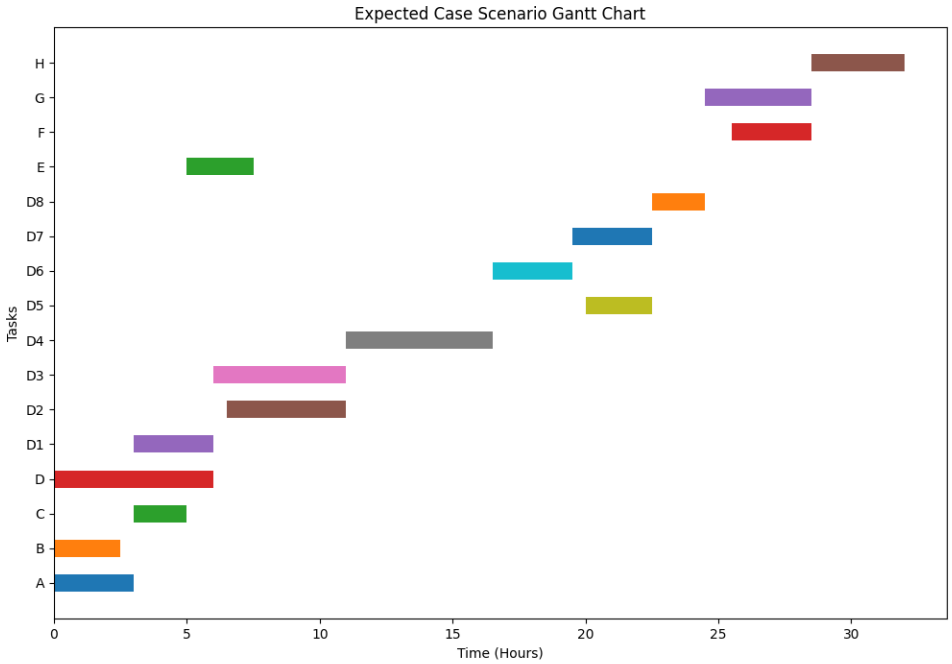
Part 4: Solution

For this part, I solved the project plan using best-case, expected, and worst-case time estimates, aiming to minimize the overall project time. The linear programming model was optimized using the CBC MILP solver, which provided results efficiently across all scenarios.

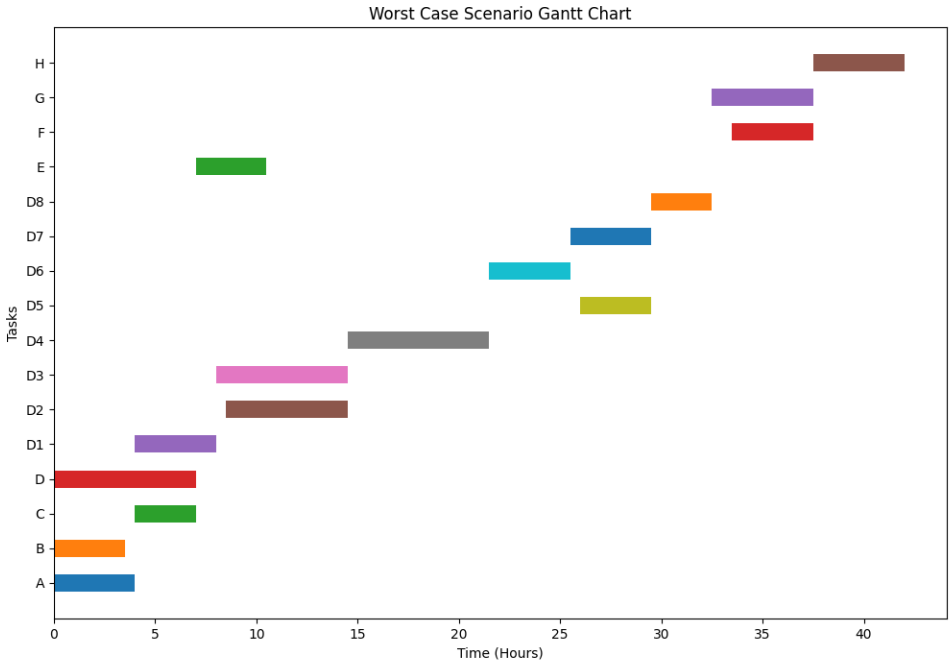
Best-Case Scenario: The solver reached an optimal objective value of 19.5, meaning the project could potentially complete in 19.5 hours under ideal conditions with no delays. This involved zero infeasibilities and took less than 0.01 seconds to solve.



Expected-Case Scenario: Here, the optimal objective was 28.5 hours, reflecting a more realistic timeframe based on average task durations. This scenario also solved with no infeasibilities in under 0.01 seconds, indicating a well-defined model.



Worst-Case Scenario: With a more pessimistic estimate, the optimal objective was 37.5 hours. This represents the longest timeline, considering possible delays in various tasks. This result was also obtained efficiently with no errors.



Each scenario's critical path was identified based on dependencies and task durations, highlighting key tasks that, if delayed, would extend the overall project timeline. The Gantt charts (also in the repository) visualize each schedule, showing how different time estimates shift project duration and critical path. These comparisons offer valuable insights into how time estimates impact project delivery and scheduling flexibility.

Part 5: Project Overview for Client

For this project, I'll develop a consumer-focused restaurant recommendation system for Marlborough, MA with regular data updates from Yelp. My role includes overall project management, data science, and system optimization at $125 per hour. Additional roles, like frontend and backend development, database management, and data engineering, are estimated at $50 per hour.

Based on the expected-case scenario of 28.5 hours for completion, the breakdown would be around $2,062.50 for my time (28.5 hours * $125) and $1,425 for other roles (28.5 hours * $50), totaling approximately $3,487.50. This excludes software and cloud hosting costs.

With a standard team size, I'd anticipate delivering a working product prototype in about 3-4 weeks (accounting for 28.5 hours spread over part-time availability and testing phases). If we added more independent contractors to the project, especially for development and testing, we could likely cut this timeline in half, delivering a prototype in around 2 weeks (by doubling team capacity and parallelizing tasks).