CS 5800 Final Project Proposal - Critical Path Method
Spring 2024
Hao Li Hejia Zhang Yuqi Hu

Background:

In the realm of project management, efficiency and timeliness are paramount. Nowadays, many of us are struggling with deadlines while procrastinating overtime. Critical Path method is a useful tool in project management to help users keep track of their progress and help them deliver their result and reach their deadline on time.Organizations across various industries strive to streamline their processes, minimize delays, and maximize resource utilization to achieve project success. One of the most effective tools in this endeavor is the Critical Path Method (CPM).

The Critical Path Method is a project management technique used to identify the sequence of tasks that must be completed on time to ensure the overall project is completed within the specified timeframe. By determining the critical path, project managers can focus their efforts on the most crucial activities, allocate resources efficiently, and mitigate potential bottlenecks that could delay project completion.

This project aims to delve deeper into the concepts, applications, and benefits of the Critical Path Method in project management. We will explore how CPM facilitates better scheduling, resource allocation and overall project optimization.

Algorithm description:

Earliest start time (ES): The earliest possible date you can start an activity considering the dependencies

Earliest finish time (EF): The earliest possible date you can complete an activity considering its ES and duration

Latest start time (LS): The last possible date you can start an activity before causing a significant project delay

Latest finish time (LF): The latest possible date you can complete a task based on its LF and duration

Task duration (t): The total amount of time it takes to complete an activity

While designing an algorithm for the Critical Path Method, it involves creating an approach to identify the longest path of planned activities to the end of a project, including the earliest and latest that each activity can start and finish without delaying the project. This process is critical for effective project management, enabling the efficient allocation of resources and time. The key component of the CRM algorithm involves several factors - tasks, durations, dependencies, and the construction of a project network diagram.

The algorithm has two parts, the forward pass and the backward pass. The forward pass is determined by using the earliest start for each activity known as ES and the earliest finish(EF). The ES of an activity equals the EF of the one before it. The EF is calculated by EF = ES + t.The backward pass assigns the last activity's EF as its latest finish. Then use the CPM formula to find the LS, which is LF – t. For the activities before that, LF is the smallest of the start times for the next activity.

Pseudocode:

Modelling the project/task as graph:

```
1. Define a Task structure with fields for name, duration, earliest start, earliest finish, latest start, latest finish, and slack.
2. Define a Graph structure to represent the project, with nodes and edges.
3. Initialize an empty graph to represent the project.
4. For each task:
    a. Create a node in the graph to represent the task.
    b. Add the task node to the graph.
5. Define dependencies between tasks as directed edges in the graph.
6. For each task in the graph:
    a. Calculate earliest start and finish times based on dependencies and task durations.
    b. Store calculated values in task objects.
7. For each task in the graph (in reverse order):
    a. Calculate latest start and finish times based on dependencies and task durations.
    b. Store calculated values in task objects.
8. For each task in the graph:
    a. Calculate slack by subtracting earliest start from latest start.
    b. Store calculated slack value in task objects.
9. Identify critical path by finding tasks with zero slack.Output the critical path.
```

Outline of the project

1. How does CPM work and what's the time and space complexity?
2. Compared to other algorithms, why does CPM perform better in project management?
3. How to apply CPM to real-life problems such as a course project?
4. What conclusions can be drawn from the study? What are the limitations and weaknesses and how can future research further our understanding on this topic?

Our Plan:

1. **Algorithm Implementation**:

    - Dive deep into the principles and concepts of the Critical Path Method (CPM). Gain a thorough understanding of how CPM works, its algorithms, and the factors influencing its effectiveness.
    - Develop a graph representation of the project, with tasks as nodes and dependencies as directed edges (more details in pseudocode above).
    - Implement algorithms to calculate earliest start and finish times, latest start and finish times, slack, and identify the critical path (more details in pseudocode above).
2. **Analysis and Evaluation**:
    - Test the implemented algorithms with various project scenarios, including different projects with different number of tasks and dependencies. Even projects with circular dependencies (which can never be completed).
    - Evaluate the performance of the algorithms in terms of time and space complexity, and accuracy in identifying critical paths.
    - Compare the performance of CPM with other project management algorithms or methods (such as Program Evaluation and Review Technique, Critical Chain Method, etc.) to find out its advantages and limitations.
3. **Conclusion and Discussion**:
    - Decide based on the results whether CPM is a good method for our project management and/or in other scenarios.
    - Discuss limitations and weaknesses that we identify during the project and suggest directions of future studies based on the project.
    - Summarize learning and future applications of the project.