Title: Personnel-to-Project Allocation Optimization Algorithm

Abstract:

The problem is which personnel are assigned to which projects. The method used to solve it is an algorithm I programmed in Python. Key findings include how different, faster, and more consistent the algorithm assigns personnel to projects than if (hypothetically) a manager were to do so by him- or herself.

Introduction:

I am addressing optimization algorithms as my topic because I wanted to get more experience with dynamic programming. For this presentation, I made a simple algorithm, but for later usage, I would like to enhance the algorithm’s capabilities for various other scenarios in “resource” allocation. Uncertainty is important in this case, because there are multiple people who can fit a project’s needs but could be better off helping in other projects. The algorithm I programmed helps address that issue efficiently.

Problem Description:

The problem is which projects should have which personnel assigned to them. The uncertainty comes from there being multiple (combinations of) people who fit the criteria for the projects. The solution is which allocation of people to which projects leads to the most efficient results. The simulated data is on information on the resources and projects respectively.

A black background with text

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

Methodology:

I programmed a “resource” allocation algorithm to solve this problem most effectively. It uses tabulation methodology from dynamic programming for accurate results.

Implementation:

I used VS Code for Python programming. The final allocation report can be seen in the app’s terminal or in a comma-separated value file.

Results:

A screenshot of a computer

AI-generated content may be incorrect.

Conclusion:

Throughout this project, I made a dynamic program for optimizing the allocation of personnel-to-projects. I learned about tabulation and other dynamic programming methodologies through doing so as well. I had to analyze them deeper in order to make my own algorithm.

References:

Since I made my own algorithm, I did not refer to other writings.