Introduction:

This report summarizes the thought process of the optimized solution and the ideas that helped me to achieve the optimized solution. My submission falls under category 3:

Initial submission always provided the optimal solutions and also included features to improve the computational performance. (Explain these features.) It is not necessary to resubmit your code for this option.

The solution I developed in my initial submission always produced the optimal solution where all test cases are passed without any issues. However, some of the test cases (last 10 test cases) took longer times (more than a day) but would produce the results.

Since no further optimization is being performed on the initial version, the code is not resubmitted.

Literature Review

After going through the project description, I initially planned to apply state-of-the-art computer science algorithm paradigms like Dynamic Programming, Greedy or Backtracking. However, I realized that Dynamic Programming is not applicable since the problem given does not contain the true sub-problems to solve optimally. After researching other techniques online, I found out that this clothing distribution problem is similar to an assignment problem in Computer Science which holds similarities to Combinatorial optimizations. These types of combinatorial optimizations can be solved by Branch-and-bound algorithms.

Hence, the algorithm I came up with is inspired from the branch-and-bound algorithm where all possible candidate solutions in the search space were explored. The candidate solution exploration path is ignored whenever a problem constraint is not satisfied. This pruning technique will optimize the time and space complexity.

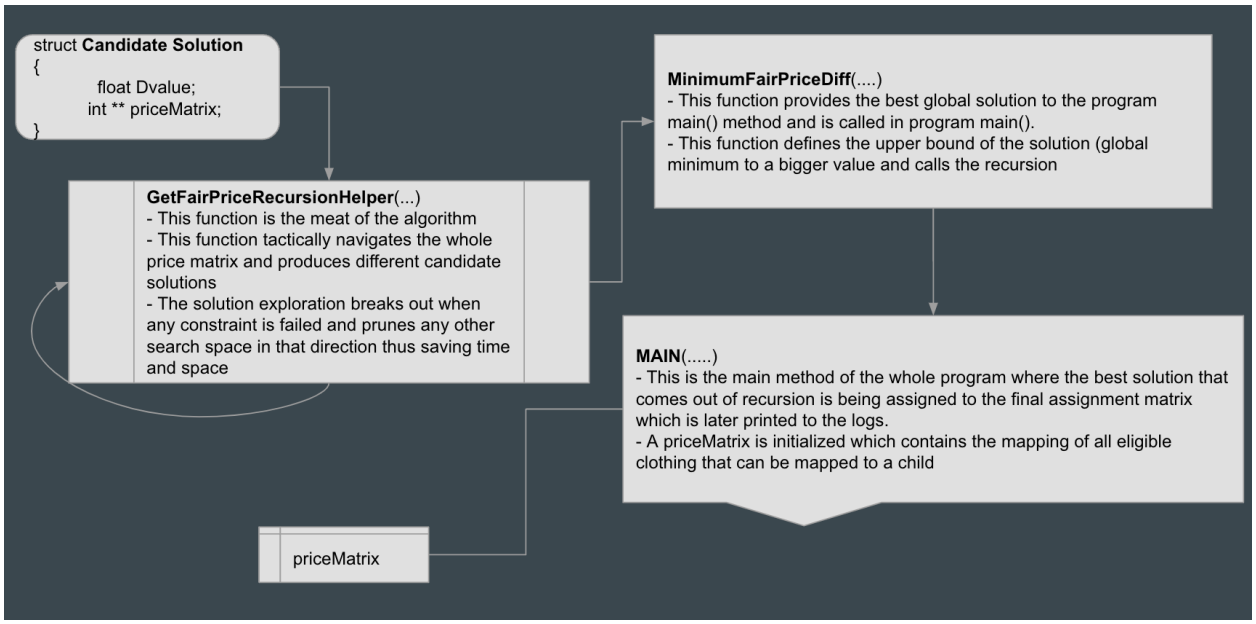
Algorithm

1. The base of the algorithm is to build all possible candidate solutions and then compare the solution distribution value with the upper bound.
2. If the value is lesser than the upper bound then overwrite the upper bound with the current value.

3. Recursion Function is being used to build a candidate solution. The last part of the recursion function computes the distribution value and compares it to the global minimum and updates the global minimum if the current distribution value is lesser.

4. The design flow of the algorithm is represented in the below diagram

1. It is important to understand the Candidate Solution data structure
2. It is important to understand the priceMatrix which is a matrix with all children mapped with all eligible clothes.
3. It is important to understand how recursion is being carried out to actually navigate through the priceMatrix and build a Candidate Solution.



Analysis:

Since we used a branch-and-bound technique, the algorithm explores all possibilities. Therefore the analysis as per the above recursion solution is:

- Time Complexity : O(numChild ^ numClothes)
- Space Complexity: O(numChild ^ numClothes)

Conclusion

As per the project requirements constraints, this is the best optimal solution I could come up with. This project gave me a great exposure on how to solve NP-Hard problems in Computer