Strategic Solutions for Warehouse Optimization: Tackling the Storage Location Assignment Problem

Ermal Belul[1,2], Dritan Nace[1] , Antoine Jouglet[1] Marwane Bouznif[2]

[1] University of Technology of Compiègne, Rue Roger Couttolenc, 60200 Compiègne, France
{ermal.belul@hds.utc.fr, dritan.nace@hds.utc.fr, antoine.jouglet@hds.utc}
[2] Savoye {ermal.belul@savoye.com, marwane.bouznif@savoye.com}

Keywords : *Storage Location Assignment Problem, Picking Cost, Replenishment Cost, Greedy Algorithm, Maximum Flow of Minimum Cost, Operations Research*

1 Introduction

A warehouse is a place where the main goal is to optimize the order preparation process. To reach that goal there are a lot of subprocesses to be optimized, one of them being storage organization[2]. Efficient storage organization hinges on resolving the Storage Location Assignment Problem. Optimizing two key properties can simplify order picking: 1) positioning frequently accessed items in easily accessible areas to minimize picking efforts, and 2) placing high-demand products in larger locations to reduce replenishment needs. The Storage Location Assignment Problem involves optimally assigning products to locations in a warehouse layout based on product dimensions, demand rates, and a picking cost function associating location accessibility with time/effort. The large real-world instances that this industry deals with on an everyday basis exceed typical academic examples in literature and these pose computational challenges. This paper introduces two main approaches that will use these inputs to generate product-to-location assignments that minimize total picking and replenishment costs.

2 Presented approaches

The first method utilizes a greedy algorithm to efficiently allocate products to storage locations while minimizing picking effort. Initially, we sort the products in descending order of frequency, prioritizing high-demand items. The algorithm iterates through each product, identifying a group of eligible storage locations based on dimensions. From this group, a unique location with the minimum location cost is selected for each product. The assigned location is removed from the available location set, ensuring each product is placed in a distinct storage spot. To account for products left unassigned due to dimension constraints, we run the greedy algorithm iteratively, starting with previously unassigned products. While this approach optimizes picking costs, it doesn't directly address replenishment costs and doesn't guarantee an optimal global solution. Nonetheless, it provides a relatively fast and practical method for finding good-quality solutions to large-scale instances of the Storage Location Assignment Problem. The complexity of this greedy approach is $O(n^2)$ due to the risk of unassignment.

Unlike our previous approach, this second solution deals with both costs, that is picking and replenishment costs. This second approach leverages network flow theory, specifically the Maximum Flow of Minimum Cost (MFM) concept, with the Hungarian algorithm as our choice for efficient and accurate execution. This method aims to maximize the flow (assigning all available products) while minimizing the total cost (including both picking and replenishment costs). The approach begins with the calculation of picking and replenishment cost matrices, considering all possible combinations of products and locations. The total cost matrix, 'C,' is

then determined as a weighted sum of these two matrices, and this matrix is provided as input to the Hungarian Algorithm. The Hungarian Algorithm, in a series of steps, manipulates the cost matrix to identify the initial feasible solution. It subtracts minimum row and column values to streamline the matrix, making it possible to find a maximal matching in a bipartite graph and optimize the assignment of products to locations[1].The Hungarian algorithm operates in $O(n^3)$ time based on the number of products and locations.

3 Results and discussion

In this section, we present the numerical outcomes obtained by applying our solution approaches to a real client's dataset, addressing the Storage Location Assignment problem. Our dataset comprises 4328 products and 5020 locations this includes a sub-domain of only the picking locations in the warehouse. The results are graphically depicted with the Picking Cost and Replenishment Cost represented on the respective axes. These results are compared to an initial client-applied solution, which was manually generated by the client. Additionally, the compilation times of each algorithm are included in the graph legends. The graphical

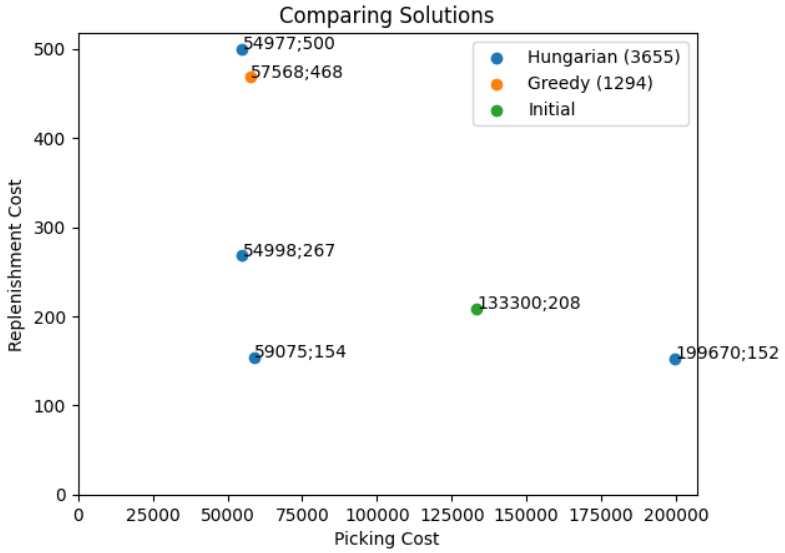

FIG. 1: Solution Results

analysis underscores a notable trait of the Hungarian algorithm: it generates a Pareto front that dominates all the other solutions. By adjusting the weights for each cost component, we can prioritize either Picking Cost or Replenishment Cost reduction, aligning the solution with specific preferences. While this demonstrates that an optimal solution can be found efficiently in theory, the large real-world instance addressed in this work poses computational challenges. Despite the polynomial complexity, computing the optimal solution still requires significant processing time as evidenced by the compilation times shown in Figure 1. The solutions developed in this work can serve as a strong starting point to explore even further how we can integrate multi-objective optimization of the Storage Assignment Problem, for example by considering the distribution of products related to the picking stations of the warehouse.

References

[1] Harold W. Kuhn. *The Hungarian method for the assignment problem. Naval Research Logistics (NRL)*, 52(1):7–21, 2005. Publisher: Wiley Online Library.

[2] Gwynne Richards. *Warehouse management: a complete guide to improving efficiency and minimizing costs in the modern warehouse.* Publisher: Kogan Page Publishers, 2017.