RMIT UNIVERSITY
School of Science, Engineering & Technology

Smart Mixed Case Palletizing Application

Team Unemployed
Nguyen Gia Khanh, Pham Trang Phi, Pavel Potemkin, Sanghwa Jung, Nguyen Quoc Hung

Academic supervisor
Dr. Khuong Nguyen Vinh

Industry supervisor
Mr. Ninh Ho Nguyen

Background and motivation

- Global e-commerce share of retail sales grew from **12%** in 2019 to **17.3%** in 2024.
- Growth in e-commerce increases demand for optimized warehousing and logistics.
- **Palletizing** is a key process for arranging similar or different-sized boxes on a pallet for storage or delivery.
- Manual palletizing is often time-consuming, error-prone, and inefficient.
- Automation can reduce human errors and improve calculation accuracy in pallet arrangement.

Objectives

- **Develop Smart Mixed Case Palletizing** (SMCP), an automated system for pallet stacking.
  - Uses algorithms to calculate optimal, stable stacking arrangements.
  - Considers box dimensions and weight for stacking decisions.
  - Aims to maximize pallet utilization and reduce manual labor.
  - Creating ABBP, a cross-platform GUI application for optimized box stacking and visualization of pallet layouts

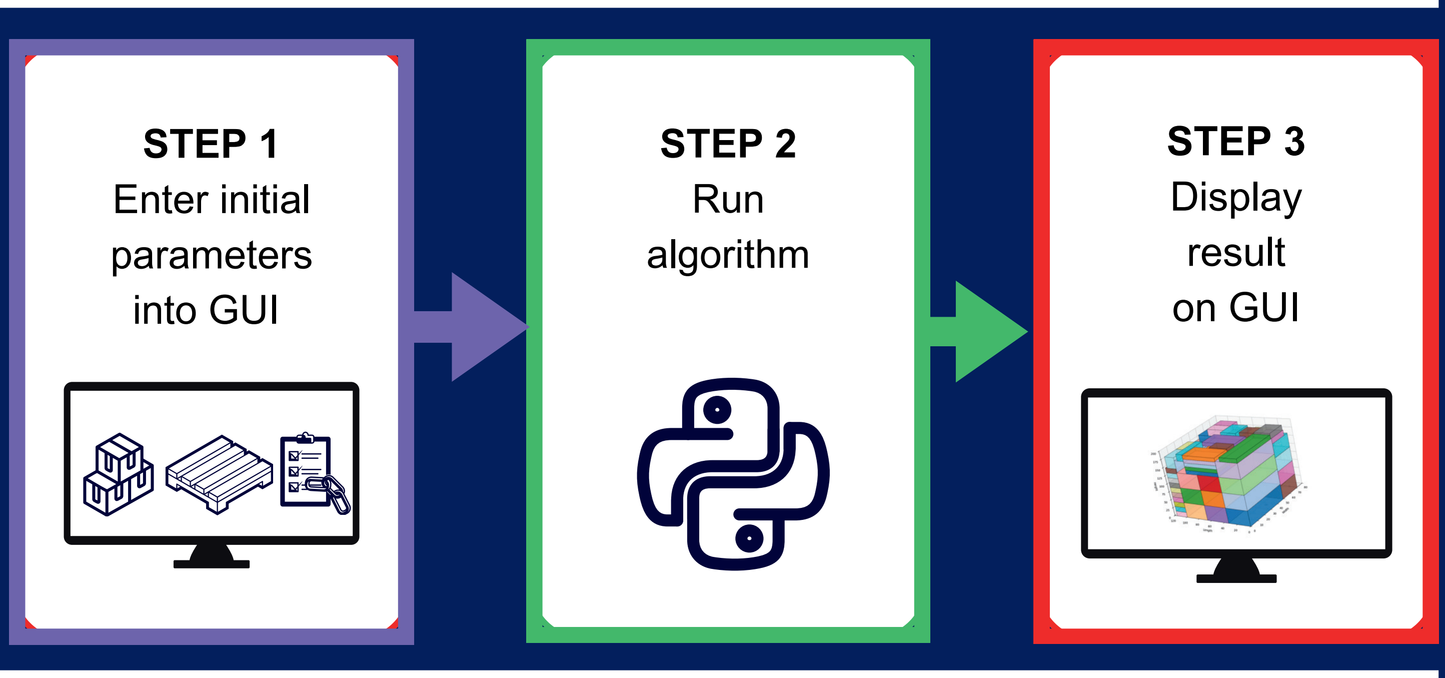
Methodology

ALGORITHM

- Built in Python using a **heuristic** place method and **genetic algorithm**.
- Applies dimension, weight, and overhang constraints.
- **Fitness score** measures compactness, optimization, and stability.

GUI

- Inputs box and pallet dimensions.
- Displays 3D pallet layout and box placement order.



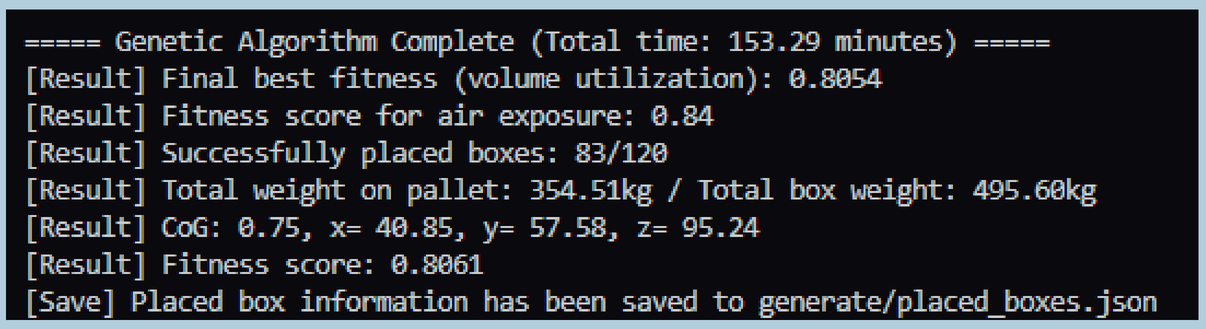
STEP 1
Enter initial parameters into GUI

STEP 2
Run algorithm

STEP 3
Display result on GUI

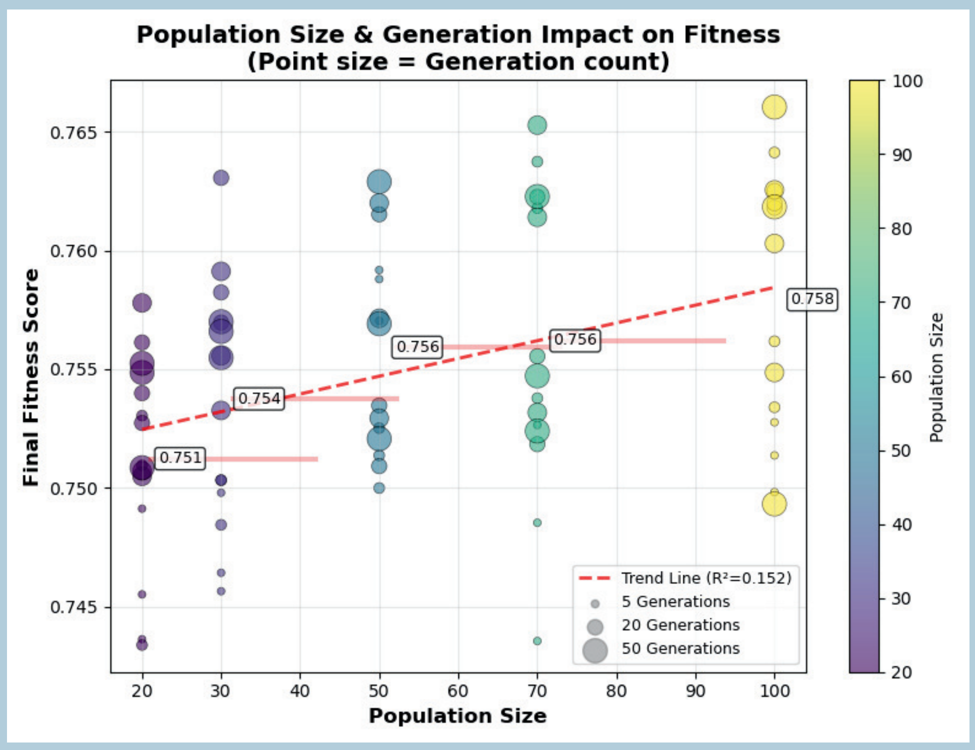
Experiments & Results

Running the algorithm with an initial population of **100 individuals** and **150 generations**, the results are displayed as follows:

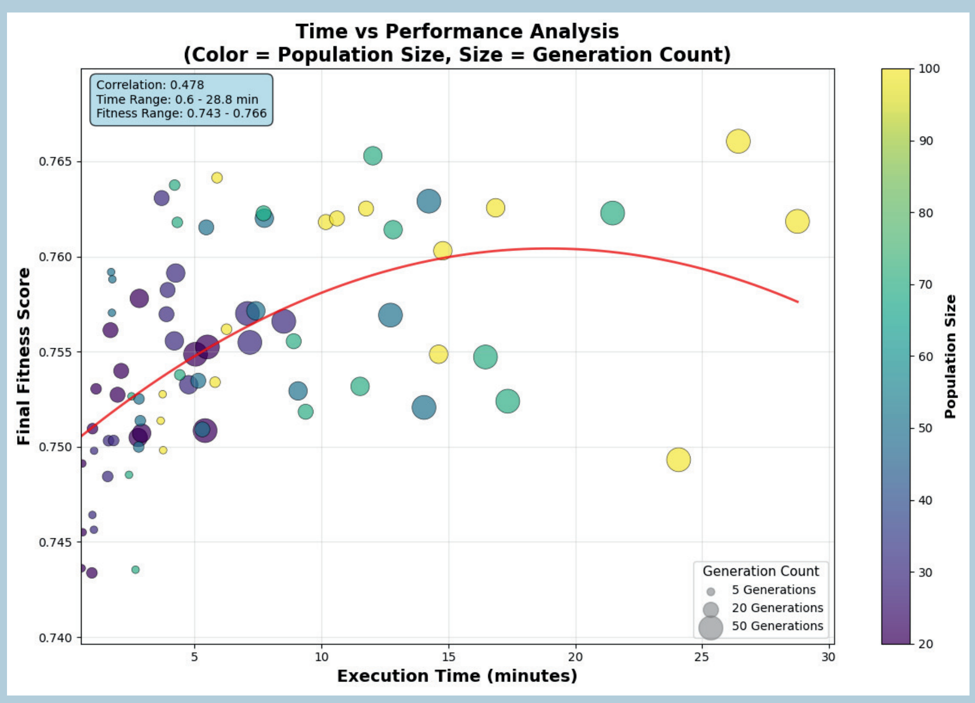


```
===== Genetic Algorithm Complete (Total time: 153.29 minutes) =====
[Result] Final best fitness (volume utilization): 0.8054
[Result] Fitness score for air exposure: 0.84
[Result] Successfully placed boxes: 83/120
[Result] Total weight on pallet: 354.51kg / Total box weight: 495.60kg
[Result] CoG: 0.75, x= 40.85, y= 57.58, z= 95.24
[Result] Fitness score: 0.8061
[Save] Placed box information has been saved to generate/placed_boxes.json
```

Running the algorithm with **varying** numbers of **generations** and **population sizes** yields the following results:

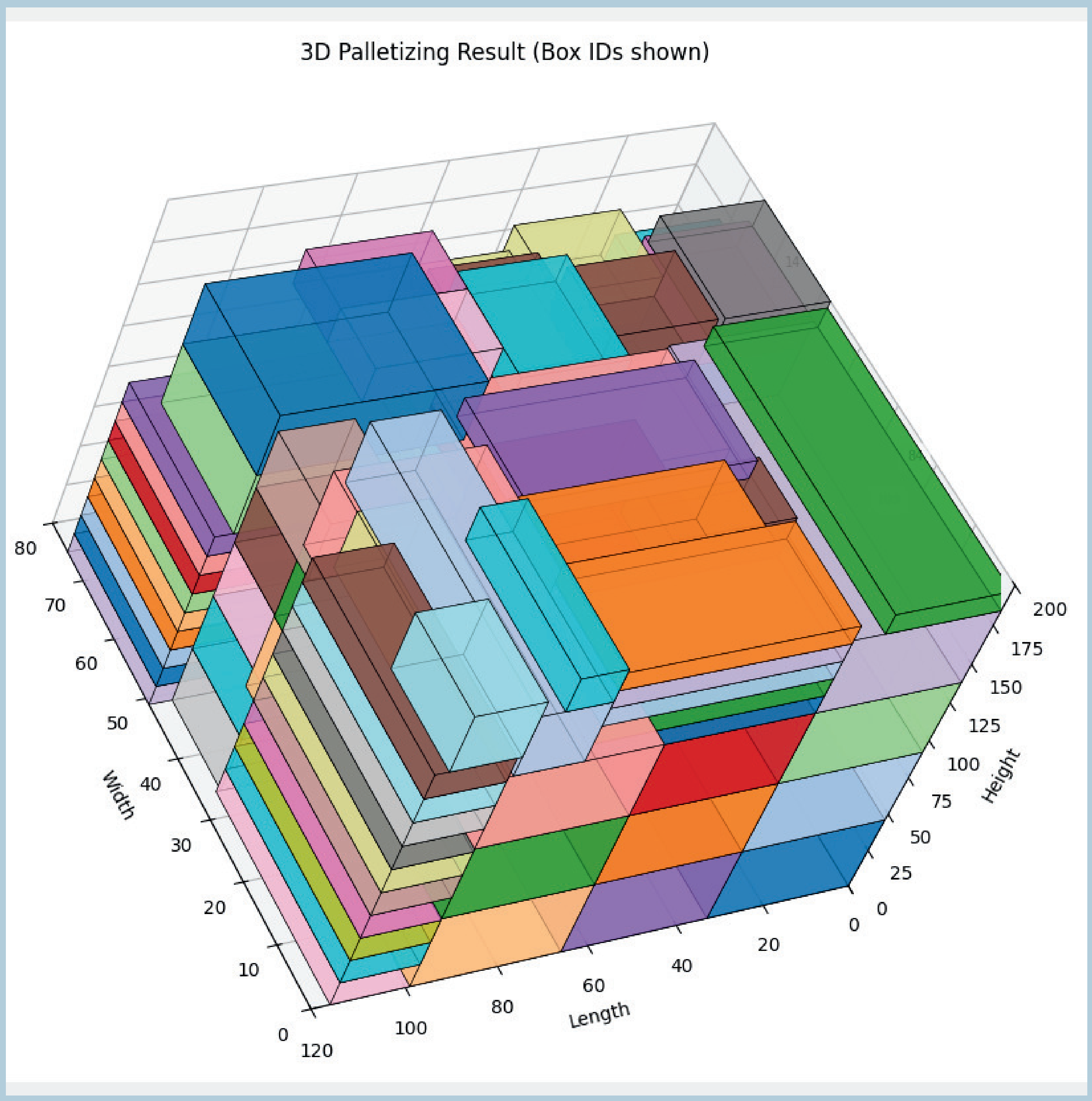


A scatter plot with Population Size on the X-axis and Final Fitness Score on the Y-axis, where point size represents Generation count.



A scatter plot with Execution Time on the X-axis and Final Fitness Score on the Y-axis, where color represents Population Size and point size represents Generation count.

3D Palletizing Results



Conclusion

The quality of the output is directly influenced by the chosen number of **generations** and **population** size. **Higher values** generally yield **better results** but require more **processing time**. According to the produced graphs, the most efficient number of generations and population size is **20** and **50** respectively. Note that the criteria for efficiency above includes fitness score and processing time. Additionally, the GUI display functions effectively, providing a clear and intuitive visualization of the outcomes.