Problem 1: The Smart Warehouse Shuffle

Domain: E-commerce Logistics

Business Story

QuickCart, a rapidly growing e-commerce company, has 15 warehouses across India. Each warehouse has 50 storage zones labeled A1-A50. Products are currently stored randomly, but the company noticed that 70% of orders contain items from just 20% of their product catalog (the “hot products”). During peak hours, workers walk an average of 12 km per shift just to fulfill orders.

The Challenge

Design an algorithm that suggests optimal product placement in the warehouse to minimize total walking distance for workers. Your solution should:

- Take historical order data (which products are frequently bought together)
- Consider that hot products change seasonally
- Account for product weight (heavy items should stay on lower shelves)
- Provide a reallocation plan that can be executed gradually (you can’t reorganize
- everything overnight)
-

Detailed Requirements

Warehouse Specifications: - 15 warehouses, each with 50 zones (A1-A50) - 5,000 unique products (SKUs) - 2,000 orders/day per warehouse during peak season - Distance between adjacent zones: 10 meters - Packing station at zone A01
Input Data: - orders.csv: 6 months of historical order data (~360,000 orders) - products.csv: Product catalog with weights, dimensions, current locations - warehouse_layout.csv: Zone coordinates and shelf specifications

Physical Constraints: - Heavy items (>15 kg) must be on LOW shelves - Medium items (5-15 kg) on LOW or MID shelves - Each zone has limited capacity (100-150 units) - Can only relocate 200 products per day
Success Metrics: - Primary: Reduce average picking distance by at least 30% - Secondary: Achieve 80% of optimal placement within 30 days - Tertiary: Balance workload across warehouse zones

Solution Approach

Recommended Algorithm: Multi-objective Genetic Algorithm
This NP-hard optimization problem requires balancing multiple objectives. A genetic algorithm is well-suited because it can: - Handle combinatorial complexity - Optimize multiple competing objectives simultaneously - Incorporate various constraint types - Find near-optimal solutions in reasonable time
Key Implementation Steps:

Phase 1: Data Analysis

•
Identify hot products using Pareto analysis (80/20 rule)
•
Build product affinity graph (frequently bought together)
•
Calculate current baseline performance
•
Create distance matrix between all zones

Phase 2: Optimization

•
Initialize population with random and greedy solutions
•
Define fitness function combining:
–

Picking distance (50%)
–
Constraint violations (30%)
–
Product affinity (20%)
•
Apply genetic operators: tournament selection, crossover, mutation
•
Use TSP approximation (nearest neighbor) for route optimization
•
Run for 100-200 generations until convergence

Phase 3: Gradual Implementation

•
Prioritize moves by impact (highest distance reduction first)•
•
•
Create daily batches of 200 products
Calculate incremental improvement after each day's moves
Adjust for operational constraints

Phase 4: Seasonal Adjustment

•
Monitor product velocity monthly
•
Re-run optimization quarterly if hot products shift significantly
•
Implement minimal changes to maintain stability

Expected Output Format

Optimal Layout Report
==========
Total Products to Relocate: 2,341 (47% of inventory)
Estimated Implementation Time: 12 days
Expected Improvements:
- Average picking distance: 8.2 km → 5.1 km (38% reduction)
- Orders per hour: 12 → 18 (50% increase)
- Worker walking time: 6.5 hrs → 4.2 hrs (35% reduction)
Day-by-Day Schedule:
Day 1 (200 products):
Move P1234 from A45 → A02 (High priority: 450 orders/month)
Move P5678 from A38 → A03 (Frequently bought with P1234)
Expected improvement today: 8% distance reduction