Shopping Time Optimization

Zeynep Meriç AŞIK
Artificial Intelligence Engineering
TOBB ETU
Ankara, Turkey

***Abstract*—This project addresses the challenge of optimizing shopping time within a market layout represented as a graph, where aisles serve as nodes containing lists of products each with associated wait times. The primary objective is to devise strategies that minimize the total shopping time for customers by intelligently navigating the graph. Two distinct approaches are proposed to tackle this problem. The first solution employs a genetic algorithm, a heuristic search method inspired by the process of natural selection, to find the optimal path through the market. The second solution formulates the problem as a Constraint Satisfaction Problem (CSP) and leverages the Gurobi optimization library to find an optimal solution. By comparing these two methodologies, the project aims to provide insights into the effectiveness of heuristic versus exact optimization techniques in complex, real-world scenarios like market navigation. The outcomes of this research could offer valuable strategies for enhancing shopping efficiency, potentially leading to applications in retail management and customer experience optimization.**

***Index Terms*—optimization, scheduling, Genetic Algorithm, Gurobi**

I. Introduction

The efficiency of shopping experiences in large retail environments is a subject of increasing importance in the realm of operational research and customer satisfaction. The layout of a market, characterized by its aisles and the distribution of products within them, plays a crucial role in determining the overall time a customer spends shopping. This project focuses on optimizing this shopping time by considering a unique set of constraints and variables such as product wait times and the physical layout of the market aisles.

In traditional shopping environments, customers often face wait times for certain products, either due to availability or service requirements. Additionally, the physical layout of the store and the distance between aisles can significantly impact the total shopping time. Addressing these factors through an optimization lens can lead to a more efficient shopping process, thereby enhancing the customer experience and potentially improving the store's operational efficiency.

This research introduces a novel approach to optimizing shopping time by modelling the market as a graph, where aisles are represented as nodes connected by edges that signify the possibility of direct passage. Each node contains a list of products along with their associated wait times. The problem is then defined as finding the optimal path through this graph that minimizes the total shopping time, considering the wait times for products and the traversal time between aisles.

To tackle this optimization problem, two distinct methodologies are employed. The first employs a genetic algorithm, a technique inspired by biological evolution, to search for optimal or near-optimal paths through the market graph. This method is well-suited to dealing with the complex, non-linear nature of the problem and offers a flexible and adaptive approach to finding solutions.

The second methodology frames the problem as a Constraint Satisfaction Problem (CSP) and solves it using the Gurobi optimization library. This approach provides a more deterministic solution path and is capable of handling a wide variety of constraints, making it suitable for finding exact solutions to the problem.

By exploring these two methodologies, this project aims to shed light on their respective strengths and weaknesses in the context of market time optimization. The findings of this research are expected to contribute valuable insights into the application of optimization techniques in retail environments, with the potential to significantly improve the efficiency of shopping processes and enhance customer satisfaction.

II. Market Layout

The market layout central to this research is conceptualized as a complex graph structure, where each aisle is represented as a node, and edges between nodes signify direct pathways connecting these aisles. This section elaborates on the intricacies of the market's layout, detailing the representation of aisles, products, wait times, and connectivity, which are critical to understanding the optimization problem at hand.

A. Aisle Representation (Nodes)

Each node within the graph symbolizes an aisle within the market. These nodes are not just placeholders; they carry significant information about the aisle, including a detailed list of products available in that aisle. Each product in this list is accompanied by a specific attribute: wait time. The wait time represents the duration a customer might need to spend in the aisle to obtain the product, accounting for factors such as service or preparation time.

B. Product Wait Times

The wait time for each product is a critical variable in the optimization problem. It varies across products and can significantly impact the overall shopping time. Products with a wait time exceeding four minutes introduce a decision-making juncture in the optimization process: should the customer wait for this product, or would it be more time-efficient to pursue other products in the meantime?

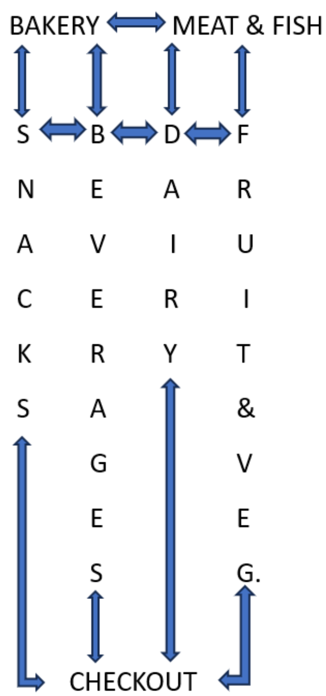

Fig. 1. Market Layout

C. Aisle Connectivity (Edges)

The edges between nodes (aisles) in the graph represent the direct walkways that customers can traverse. Aisles can be listed as entrance, snacks, beverages, dairy, meat and fish, fruits and vegetables, bakery, and lastly checkout. Entrance and checkout are treated as aisles (nodes) to connect to the aisles, but they do not own items so their item lists are empty unlike any other aisle.

Not all aisles are directly connected. For example, customer cannot go from entrance to bakery aisle directly. This aspect of the layout adds a layer of complexity to the optimization problem, as it restricts the direct paths a customer can take, thereby influencing the optimal route and the overall shopping time. Figure above shows the market layout and the connections between aisles (Fig. 1).

D. Traversal Cost

A uniform traversal cost is associated with moving from one aisle to another, set at two minutes. This cost is a simplification to standardize the time it takes to pass through an aisle, irrespective of the actual physical distance. This assumption is crucial for the model to capture the essence of the time penalty incurred when moving between different sections of the market.

To sum up; the market layout, with its aisles represented as nodes, connected by edges, and each containing a list of products with associated wait times, forms the backbone of the optimization problem. This graph-based representation allows for the application of sophisticated optimization techniques, such as genetic algorithms and CSP solvers, to navigate the complexities of the market layout efficiently. The goal is to determine the optimal path that minimizes total shopping time, considering both the wait times for individual products and the cost associated with moving through the market.

III. Genetic Algorithm

The genetic algorithm (GA) is a bio-inspired optimization technique that mimics the process of natural selection and genetics to solve complex problems. In the context of optimizing shopping time within the market layout described earlier, the GA provides a flexible and adaptive approach to finding efficient paths through the market, considering product wait times and aisle traversal costs. This section outlines the implementation of the genetic algorithm for this specific problem.

A. Initialization

The GA begins with the generation of an initial population of solutions, where each solution, or ”individual,” represents a possible path through the market. Each path is encoded as a sequence of nodes (aisles) that the customer will visit. The initial population is typically generated randomly, ensuring a diverse set of starting points for the optimization process. Population size is kept on 50 and number of generations produced on 100.

B. Fitness Evaluation

The fitness of each individual in the population is evaluated based on how well it solves the optimization problem. In this case, the fitness function calculates the total shopping time for a given path, including the wait times for products in each aisle and the traversal costs between aisles. While doing so, the function uses Breadth-First-Search (BFS) to find the shortest path between two aisles (as each aisle takes 2 minutes to traverse, optimizing the sub-paths is important) and uses an ordered items set not to waste time while waiting the product to be ready to pick up. This ordered items set includes the items with longer than four minutes of waiting time. When an item is in this set, a minute passes for ordering the item and the customer can come back to pick it up any time after it is ready (the wait time has passed since ordering). Meanwhile, all the costs between ordering and picking up are counted within total time. The objective is to minimize this total shopping time, so lower fitness values are preferable.

C. Selection

Selection is the process of choosing individuals from the current population to breed a new generation. Various selection methods exist, but they all aim to preferentially select individuals with better fitness, simulating natural selection.

D. Crossover

Crossover, or recombination, is a genetic operator used to combine the genetic information of two parents to generate new offspring. For the market optimization problem, crossover involves swapping segments of the parents’ paths to create new paths. Precautions are taken to ensure that the resulting paths are valid and do not, for example, include duplicate visits to the same aisle unless explicitly allowed by the problem constraints (such as products with long wait times). The crossover rate used for this problem is 0.8.

E. Mutation

Mutation introduces random changes to the offspring's genetic material to maintain genetic diversity within the population. In the context of this problem, mutation might involve randomly altering an aisle in a path or swapping two aisles in the sequence. The mutation rate is typically kept low with 0.1 to prevent excessive randomization of solutions.

F. Termination

The algorithm iterates through these steps, generating new populations until a termination condition is met. Termination criteria can include reaching a maximum number of generations, achieving a solution of acceptable fitness (picking up all items and heading checkout), or observing minimal improvement over several generations.

By iteratively evolving solutions and applying the principles of selection, crossover, and mutation, the genetic algorithm navigates the solution space of the market layout optimization problem. This approach is particularly well-suited to the problem's complex, non-linear nature and can efficiently search for near-optimal paths that minimize the total shopping time, considering both product wait times and aisle traversal costs.

IV. CSP Solver

The Constraint Satisfaction Problem (CSP) approach to optimizing shopping time in the market layout involves formulating the problem in a manner where the solution satisfies a set of constraints. This method is deterministic and seeks an exact solution using the Gurobi optimization library through its Python interface, Gurobipy. This section details the process of applying CSP principles with Gurobipy to find the optimal shopping path.

A. Problem Formulation

The first step is to translate the market layout and shopping objectives into a mathematical model. This involves defining parameters, variables, constraints, and the objective function within the context of the market graph, where aisles are nodes and edges represent possible paths between these aisles.

Parameters can be listed as product information and aisles. Aisles are the same as the ones given above. Products are read from a dataset that has product name as ID, waiting time, price and aisles as columns. Even though price has nothing to do with the algorithm it is still calculated to give the user a realistic experience.

Each variable represents a decision point in the shopping path, such as choosing to visit an aisle or deciding the sequence of aisle visits. This model uses two variables for each aisle. First one is a binary variable that decides whether an aisle is visited or not. Second one is also binary but has two parameters - both aisles, which determines whether the customer should use the path between the two aisles or not.

The model includes constraints to ensure valid paths through the market. These involve: Ensuring that if an aisle is chosen, all prerequisites (like prior visits to specific aisles for product availability) are met. Enforcing the connectivity between aisles, ensuring that the path from one aisle to the next is possible given the market layout. Incorporating product wait times, ensuring that if a product's wait time exceeds four minutes, the path accounts for this either by waiting or by optimizing the time to pick up other items.

B. Objective Function

The objective function quantifies the goal of minimizing the total shopping time, incorporating both the wait times for products in each visited aisle and the traversal times between aisles. The function is formulated to sum these time components across the chosen path, providing a total time that the CSP solution aims to minimize.

After defining these particles, Gurobi employs advanced mathematical programming techniques to find the optimal solution within the defined constraints. Upon completion, the solution provided by Gurobi includes the optimal path through the market, minimizing the total shopping time. The path is derived from the values of the decision variables in the optimized model, indicating which aisles to visit and in what sequence to minimize time spent shopping, considering wait and traversal times.

The CSP approach with Gurobipy offers a structured and precise method to tackle the market layout optimization problem. By defining the problem in terms of variables, constraints, and an objective function, and then leveraging the powerful Gurobi optimizer, this method provides an exact solution that guarantees the minimization of shopping time within the given constraints. This approach is particularly useful for scenarios where an exact solution is preferred over the heuristic solutions provided by methods like genetic algorithms.

V. Comparison and Results of the Algorithms

In comparing the results obtained from the genetic algorithm (GA) approach and the Constraint Satisfaction Problem (CSP) approach solved using Gurobi, the evaluation of their performance is focused on several key metrics: solution optimality, runtime, and the robustness of each method under varying market conditions. This comparative analysis provides insights into the strengths and limitations of each method when applied to the market layout optimization problem.

A. Solution Optimality

The GA approach tended to find near-optimal solutions due to its heuristic nature. It excelled in scenarios with highly complex market layouts and a vast number of products, where the search space was too large for exhaustive exploration. The solutions were generally very good, but there was no guarantee they were the absolute best possible and there is still no guarantee if the same algorithm with the same fitness function would give the best or at least a good solution for a different market layout.

The CSP approach, on the other hand, provided exact solutions by precisely defining the problem's constraints and objectives. When the market layout and problem size were manageable, Gurobipy was able to find the optimal path, ensuring the minimum possible shopping time given the constraints. But when the connections between aisles get more complex, it might not be able to since the logic behind this algorithm is simpler than GA.

B. Runtime

The runtime of the GA varied significantly with the size of the initial population and the number of generations. For larger problems, the GA could become computationally expensive, though it remained relatively efficient for most test cases by converging to satisfactory solutions within a reasonable number of generations.

The CSP solution's runtime was highly dependent on the complexity of the market layout and the tightness of the constraints. For this problem, Gurobipy solved the CSP very quickly. However, as the problem size and complexity increases, the runtime could grow significantly, sometimes making it impractical for very large or highly complex market layouts. The runtime approximate cannot be made in seconds since it highly relies on the size of the input shopping list.

C. Robustness and Scalability

The GA demonstrated high robustness and adaptability to various market configurations and sizes. Its performance degraded gracefully with increasing complexity, making it a versatile tool for a wide range of problem sizes, though with less predictability in solution quality for extremely complex scenarios.

The CSP approach was highly robust for problems within its computational reach, consistently delivering optimal solutions. However, its scalability was limited by computational resources, especially for very large and complex market layouts where the number of constraints and variables could explode, leading to longer solve times or infeasibility.

VI. Conclusion

The comparison reveals that both the GA and CSP approaches have their niches where they excel. The GA offers a flexible, heuristic-based method that is particularly useful for large, complex problems where an exact solution is less critical than finding a good solution quickly. It is best suited for scenarios where runtime efficiency and solution quality balance are essential. On the other hand, the CSP approach, when solvable within practical time frames, guarantees optimality and is preferable for smaller to medium-sized problems or when the absolute best solution is necessary, despite potentially longer run times for complex scenarios.

References

[1] Jia Luo (LAAS-CDA), Didier El Baz (LAAS), A Survey on Parallel Genetic Algorithms for Shop Scheduling Problems

[2] Sayedmohammadreza Vaghefinezhad, Kuan Yew Wong, A Genetic Algorithm Approach for Solving a Flexible Job Shop Scheduling Problem

[3] Maissa Irmouli, Nourelhouda Benazzoug, Alaa Dania Adimi, Fatma Zohra Rezkellah, Imane Hamzaoui, Thanina Hamitouche, Malika Bessedik, Fatima Si Tayeb, Genetic Algorithm enhanced by Deep Reinforcement Learning in parent selection mechanism and mutation : Minimizing makespan in permutation flow shop scheduling problems

[4] Alec Kirkley, Inference of dynamic hypergraph representations in temporal interaction data

[5] Haoqi He, Quantum Annealing and Graph Neural Networks for Solving TSP with QUBO

[6] https://colab.research.google.com/github/Gurobi/modeling-examples/blob/master/milp_tutorial/introduction_to_modeling.ipynb