Genetic Algorithm for the Knapsack Problem and the Traveling Salesman Problem

Group 29: Egor Sementsul (i6290310), Husam Mekhallalati (i6215229)

March 17, 2024

Contents

1 Implementation and Design Choices

In our implementation, the genetic algorithm (GA) was adapted to optimize solutions for the knapsack and traveling salesman problems (TSP). This framework is designed to be generic, enabling the selection from a variety of crossover, mutation, and selection strategies to best fit the problem at hand. Population initialization and fitness evaluation methods are specifically defined for each problem, reflecting their unique requirements and constraints. This modular design allows for easy adaptation and optimization across different problem domains.

Knapsack Problem
In the Knapsack GA, our population consists of binary strings representing potential item selections for the knapsack, with each bit corresponding to the inclusion (1) or exclusion (0) of an item. The fitness of an individual reflects the cumulative value of the selected items in a solution, serving as a utility measure that our algorithm seeks to maximize. This total is calculated under the constraint that the cumulative weight does not surpass the knapsack's limit. Solutions that exceed the weight capacity are penalized (fitness = 0), ensuring that the GA evolves only valid combinations of items.

Traveling Salesman Problem
For the TSP, we define the population as a set of permutations, with each representing a potential route. Each individual permutation corresponds to a sequence of city visits. Fitness is determined by the inverse of the route's total distance, with a smaller negative

value indicating a shorter route. Thus, maximizing the fitness function leads to the minimization of the total travel distance. To maintain valid solutions, we implemented specialized crossover and mutation strategies which respect the uniqueness of each city in a tour, avoiding repeated visits to a single city, and ensuring each city is visited exactly once before returning to the starting point.

Selection Strategies

In our genetic algorithm, two selection strategies facilitate diversity and robustness in evolving populations. The **roulette wheel** selection probabilistically favors individuals based on their fitness proportion to the total, offering a chance even to less fit individuals and thus preserving genetic diversity. The **tournament selection**, on the other hand, introduces a competitive edge by only selecting the best out of a randomly chosen subset, promoting stronger traits more aggressively. Both roulette wheel and tournament selection are implemented due to their complementary strengths. The roulette wheel approach ensures all individuals have a chance to be selected, promoting genetic variation. Tournament selection, by choosing the best from a random subset, ensures the propagation of the fittest individuals. This caters well to the needs of both knapsack and TSP problems by maintaining a diverse gene pool while still encouraging the survival and reproduction of the strongest candidates, thus enhancing the overall performance of the genetic algorithm.

Crossover Strategies

Our genetic algorithm for the Knapsack problem utilizes two crossover strategies: single-point and uniform crossovers. The **single-point crossover** combines parent solutions at a randomly chosen point, merging their characteristics up to that point and potentially yielding offspring with a balanced mix of both parents' traits. **Uniform crossover**, in contrast, randomly exchanges genes between parents throughout the entire length of the solution, promoting diverse offspring.

For the TSP, ensuring solution validity (each city is visited once and only once) is crucial. The ordered, partially mapped (PM), and cycle crossovers are implemented to maintain this requirement. **Ordered crossover** maintains a sequence of cities from one parent while filling the rest with non-duplicated cities from the other parent. **PM crossover** swaps segments between parent solutions and employs a gene mapping strategy to correct any resulting duplicates, preserving the integrity of city sequences. **Cycle crossover** identifies cycles between two parent solutions, providing offspring that are a mix of both parents while keeping each city's position unique. These crossover methods are pivotal for generating valid TSP routes.

Mutation Strategies

For the Knapsack problem, the **bit-flip mutation** strategy is utilized, where each gene in an individual's binary string may be inverted from 0 to 1, or vice versa, according to a predetermined mutation rate. This method introduces variability within the population, fostering exploration of the solution space.

In addressing the TSP, three mutation strategies are employed to ensure route validity:

1. **Insertion Mutation** selects a city and reinserts it into a different position within the route, altering the tour structure while maintaining all cities.

2. **Inversion Mutation** involves selecting a segment of the route and reversing the order of cities within it, which can reveal shorter, previously unexplored paths.

3. **Swap Mutation** exchanges the positions of two cities, a mechanism for local tour adjustments.

These mutation techniques are instrumental in introducing solution variations, enabling the genetic algorithm to explore an extensive range of possibilities and evade premature convergence on suboptimal tours.

2 Research Questions

We investigate the following key aspects through our experiments:

1. **Impact of Selection Strategy:** Evaluating how roulette wheel and tournament selection strategies influence the GA's efficiency and solution quality.

2. **Crossover Strategy Impact:** Analyzing the effect of various crossover strategies on the convergence and solution quality of the GA.

3. **Mutation Rate Influence:** Understanding how different mutation rates impact the GA's performance and its ability to find optimal solutions.

3 Experiments and Results

Before delving into the specific experiments, it's important to outline the general experimental setup. All experiments were conducted with a population size of 100 and spanned over 200 generations, employing a crossover rate of 0.8. By deciding to perform crossover only with 80% we allow some individuals to pass unchanged into the next generation. This approach gives the population more room to explore the solution space therefore avoid premature convergence. For the knapsack problem, the items were defined with a maximum weight limit of 80 and the number of items was set to 20, while for the TSP, a configuration of 50 cities on a circular layout was used.

Impact of Selection Strategy
The first set of experiments evaluated the impact of roulette wheel and tournament selection strategies. For both the knapsack and TSP problems, tournament selection consistently led to higher average fitness values, suggesting its superiority in guiding the population towards optimal solutions. The experiments utilized uniform crossover and bit-flip mutation for the knapsack, and ordered crossover and inversion mutation for the TSP.

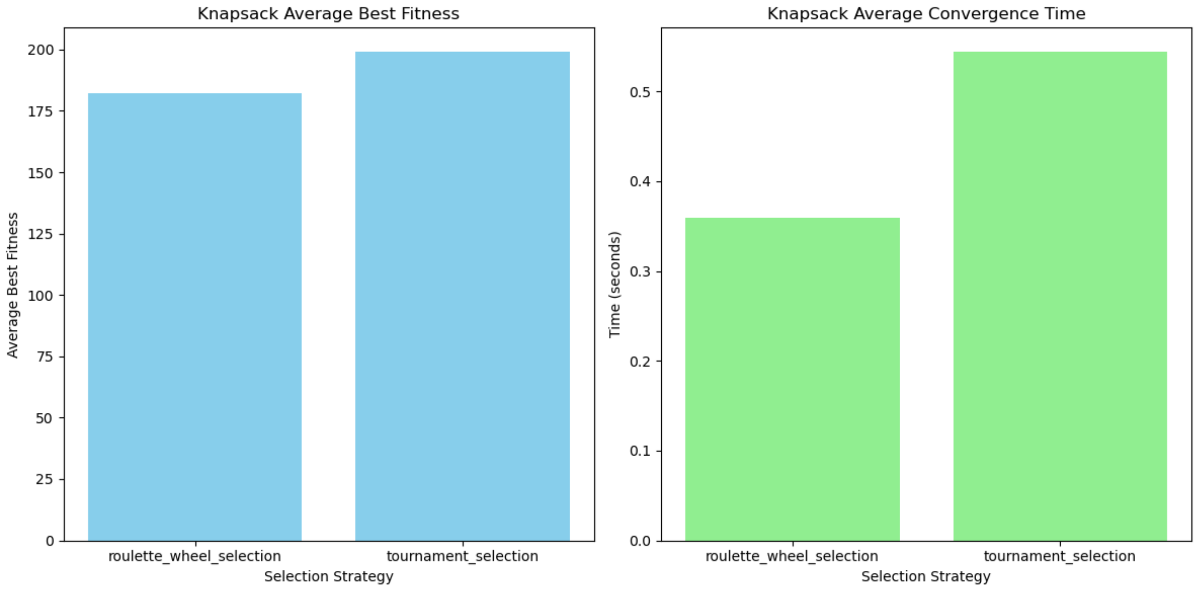

Figure 1: Impact of selection strategies on the knapsack problem.

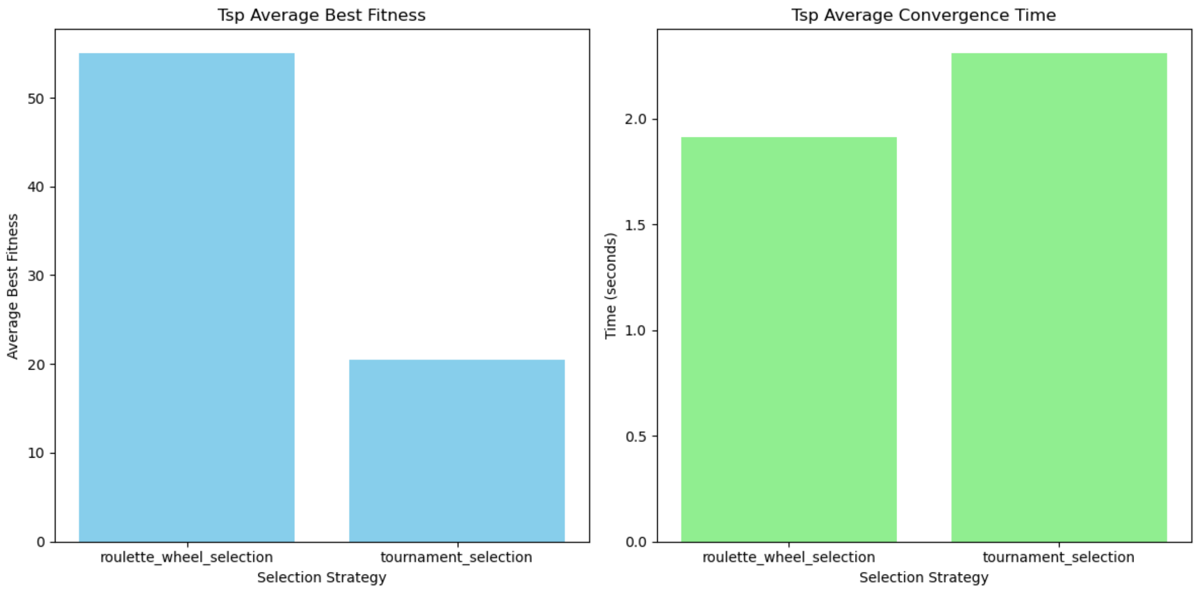

Figure 2: Impact of selection strategies on the TSP problem.

Crossover Strategy Impact

This section focuses on the influence of different crossover strategies. The knapsack problem experiments revealed minimal differences in performance between single-point and uniform crossover. For the TSP, ordered crossover emerged as the most effective in producing quality solutions. The selection strategy used was tournament selection, with bit-flip mutation for the knapsack and inversion mutation for the TSP.

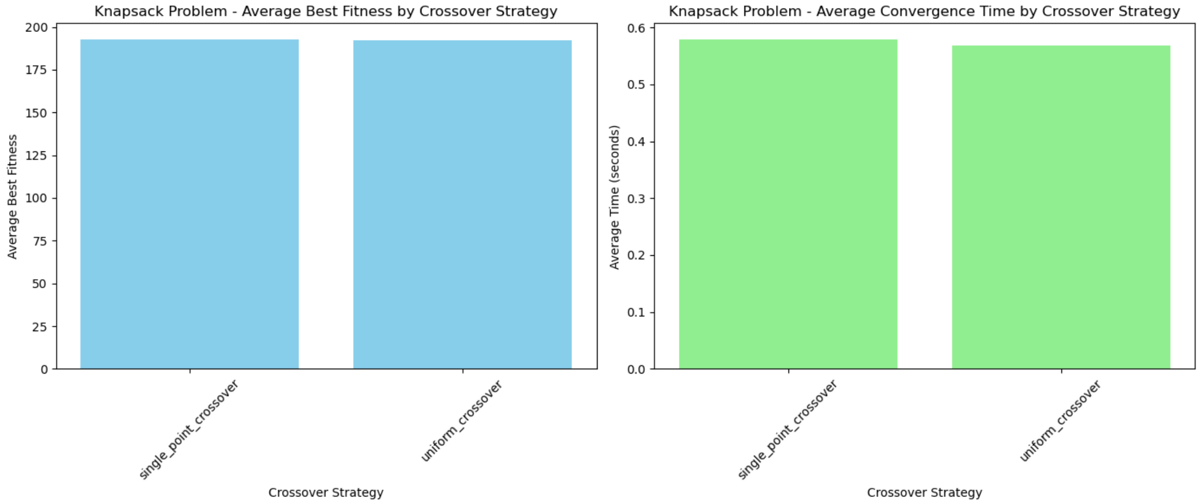

Figure 3: Comparison of crossover strategies for the knapsack problem.

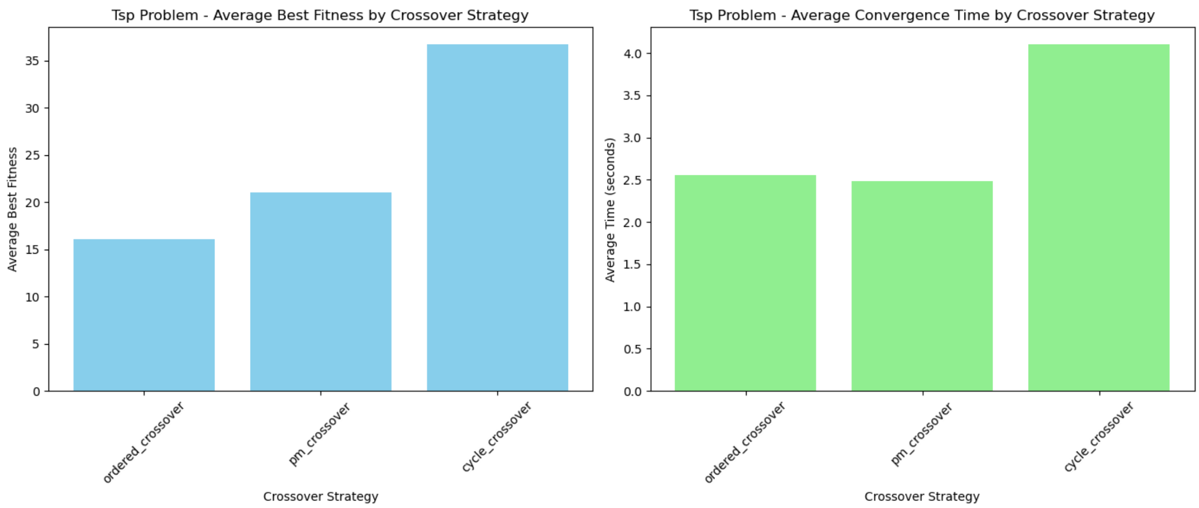

Figure 4: Comparison of crossover strategies for the TSP problem.

Mutation Rate Influence

Investigating mutation rates revealed distinct optimal settings for each problem. For the knapsack, lower mutation rates led to superior fitness, peaking at 0.01, highlighting the importance of genetic stability. In contrast, the TSP benefited from higher rates, with 0.125 fostering better exploration and solution quality. These experiments, conducted with tournament selection and uniform crossover for knapsack and ordered crossover for TSP. Bit-flip mutation for the knapsack problem and inversion mutation for the TSP was used, to maintain consistency.

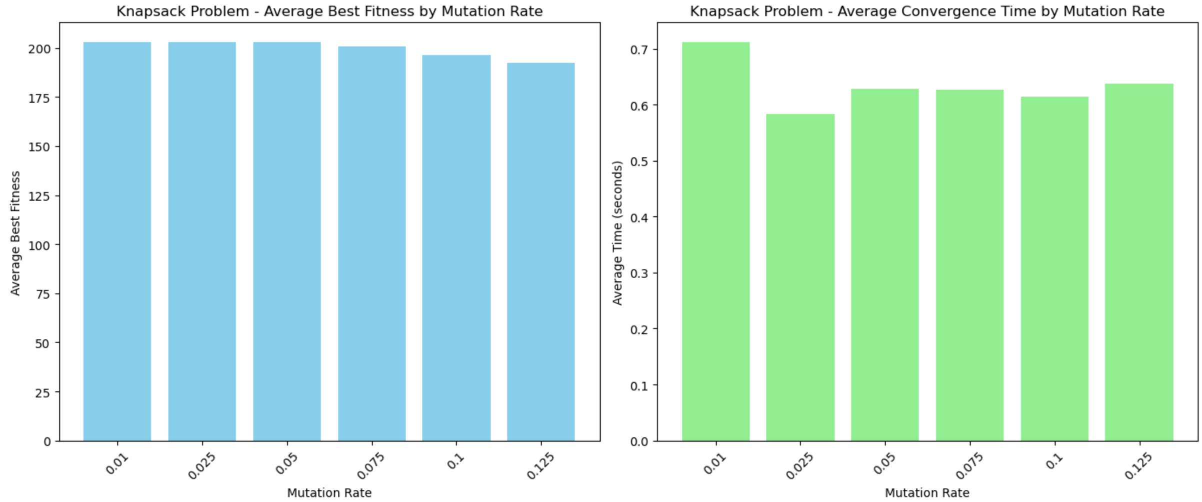

Figure 5: Influence of mutation rate on the knapsack problem.

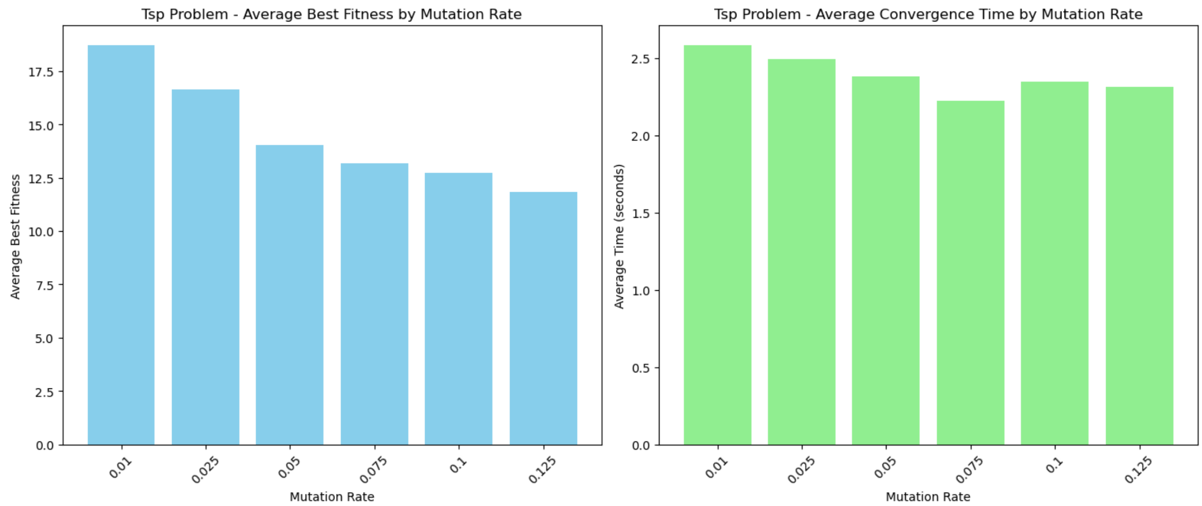

Figure 6: Influence of mutation rate on the TSP problem.

4 Discussion and Conclusion

Our experiments across different configurations of the genetic algorithm for solving the knapsack and TSP problems revealed insightful trends regarding selection and crossover strategies, as well as the influence of mutation rates.

For the knapsack problem, tournament selection demonstrated superior performance over roulette wheel selection in terms of achieving higher average fitness, suggesting its effectiveness in maintaining selective pressure towards optimal solutions. The crossover strategies showed comparable results, with a slight edge for the uniform crossover in terms of fitness, though the differences were marginal. Mutation rate analysis highlighted a preference for lower rates, indicating the importance of preserving advantageous genetic material.

In contrast, the TSP problem benefited significantly from tournament selection and the ordered crossover strategy, which outperformed others in terms of average fitness. This underscores the importance of maintaining solution validity and effective gene recombination for complex route optimization problems. Higher mutation rates proved beneficial for the TSP, emphasizing the need for exploration in a more complex solution space.

The average convergence times provide further insights into the efficiency of the genetic algorithm across different problem settings. For the knapsack problem, tournament selection exhibited longer convergence times compared to roulette wheel selection, implying a more extensive search process. Conversely, for the TSP, convergence times were consistently longer across all strategies, with tournament selection and more complex crossovers like cycle crossover taking the longest. This highlights the intrinsic computational complexity associated with solving the TSP, where more sophisticated strategies necessitate additional computational time to navigate the solution space effectively.

In conclusion, the optimal setup for the knapsack problem in our experiments involves using tournament selection, uniform crossover, and a lower mutation rate, balancing the exploitation of high-quality solutions with sufficient exploration. For the TSP, tournament selection, ordered crossover, and a higher mutation rate form the best configuration, emphasizing the need for robust exploration mechanisms to navigate the intricate solution landscape. These findings highlight the adaptability of genetic algorithms to diverse problem domains, with the selection of appropriate operators and parameters being crucial for achieving optimal performance.

5 Use of LLMs

Within the scope of this lab, ChatGPT was utilized during both the implementation and report writing phases. During the implementation, GPT acted as an advisor, guiding us through various design choices. Specifically, we consulted ChatGPT about which crossover and mutation strategies would ensure solution validity in the TSP setup. Additionally, ChatGPT suggested a high-level code structure designed to maintain a generic approach, facilitating the application to a wide range of problems with minimal modifications. Regarding the report, the final version was reviewed by ChatGPT to proofread and rectify any grammatical or language inaccuracies.