CS3CI Computational Intelligence: Cutting Stock Problem – The Effect of Immigration Method and Adaptive Mutation Rate

Faran Zafar,

Aston University

CS3CI Computational Intelligence: Cutting Stock Problem

Contents

Table of Figures

Introduction:

The Cutting Stock Problem (CSP) is a practical challenge faced by industries like manufacturing and logistics. It's about finding the best way to cut larger materials into smaller pieces with minimal waste. Solving CSP efficiently is crucial because it helps reduce waste, save costs, and increase material efficiency, making it a key factor in enhancing operational effectiveness in various sectors.

Literature Review:

The CSP has been a subject of research since 1939, formulated initially by Kantorovich. Traditional approaches involved linear programming, focusing on material optimization during cutting stages. Recent advancements have seen computational intelligence being applied to CSP. Techniques like genetic algorithms and neural networks offer adaptable solutions, particularly in optimizing cutting patterns and predicting optimal solutions from historical data.

Proposed Computational Intelligence Solution:

Introduction to My Solution:

In addressing the CSP, I chose to use an evolutionary algorithm, specifically a genetic algorithm, due to its flexibility and robustness in exploring and optimizing complex problem spaces.

Detailed Description:

Baseline:

Assumptions:

We assume that we have unlimited supply of stocks to fulfill the order quantities

Initialization:

The initial population is created with diverse cutting patterns. This is achieved using the **initialize_population** function, which generates a variety of feasible cutting configurations.

Fitness Calculation:

The fitness of each solution is calculated based on cost, waste, and penalties for unmet quantities. The **fitness** function evaluates these aspects.

Selection Process:

I use tournament selection to select the best individuals for cross over and mutation method thus enabling the survival of the fittest.

Algorithm Workflow:

The **genetic algorithm** function orchestrates the entire process, managing the population through generations until a suitable solution is obtained.

Novelty:

Adaptive Mutation Rate:

To adapt to the problem's complexity over generations, I used an adaptive mutation rate, adjusting the mutation likelihood as the algorithm progresses.

Penalty Mechanism in Fitness Function:

The fitness function includes a penalty mechanism for not meeting quantity requirements and encourages diversity in stock use, ensuring a balance between resource utilization and solution optimality.

Immigration Method:

This method introduces a completely random individual to population at certain intervals of the generation to introduce more diversity to the population

Experimental Methodology:

1. **Objectives and Hypotheses**:

“The implementation of a genetic algorithm with adaptive mutation rates and immigration method significantly reduce the total cost of stock usage in the Cutting Stock Problem compared to traditional methods or basic genetic algorithms without these enhancements.”

1. **Objective**: Test the effectiveness of a genetic algorithm against Random Selection Algorithm
2. **Objective**: Investigate the Role of Adaptive Mutation in the EA
3. **Objective**: Evaluate the effect of Immigration Method on EA

2. **Experiment Design**

Sample Size: 100

1. **Experiment - Algorithm Effectiveness**:

For this Experiment I'm comparing my EA Algorithm solution with a baseline random algorithm to see how does EA outperform when it comes to problems such as CSP on metrics such as cost and computational power.

- **Null Hypothesis (H0)**: There is no significant difference in performance between the Random Algorithm and the Evolutionary Algorithm.
- **Alternative Hypothesis (H1)**: There is a significant difference in performance between the Random Algorithm and the Evolutionary Algorithm.
- **RA: Parameters:** 100 Generations, 2 Seconds
- **EA: Initial Parameters:** 100 Generations, Population Size:10, Initial Mutation Rate: 0.08, Tournament Size: 10, 2 Seconds

2. **Experiment – Effect of immigration method to break from Local Optima**:

In this experiment I tried to break the local optima of my population by introducing random individual in each generation and see how it effects it. Adding diversity into population thus stopping them from having the same individual in every generation

- **EA Without Immigration: Initial Parameters:** 10 Generations, Population Size:100, Initial Mutation Rate: 0.08, Tournament Size: 10
- **EA With Immigration Method: Initial Parameters:** 10 Generations, Population Size:100, Initial Mutation Rate: 0.08, Tournament Size: 10, Immigration Size: 5, Immigration Frequency: 2

3. **Experiment: Test the Effect of Adaptive Mutation Rate**

This Experiment is designed to test the effect of adaptive mutation on the population. How it effects the algorithm to look for more optimum solution in the search space

- **Initial Parameters:** 200 Generations, Population Size:1000, Initial Mutation Rate: 0.05, Tournament Size: 500

Analysis of Experimental Results:

Experiment 1:

T-Test Results

- T-statistic: 21.2703
- P-value: Approximately 6.67e-38

Random Search:

Statistic	Performance Metric
Mean	1947.37
Median	1944.50
Minimum	1937.00
Maximum	1965.50

Table 1: Showing Results of Random Search for CSP

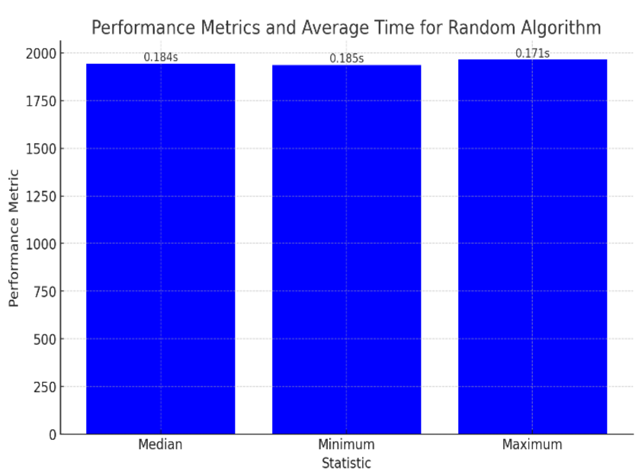

Figure 1: Block Graph showing Cost achieved in time

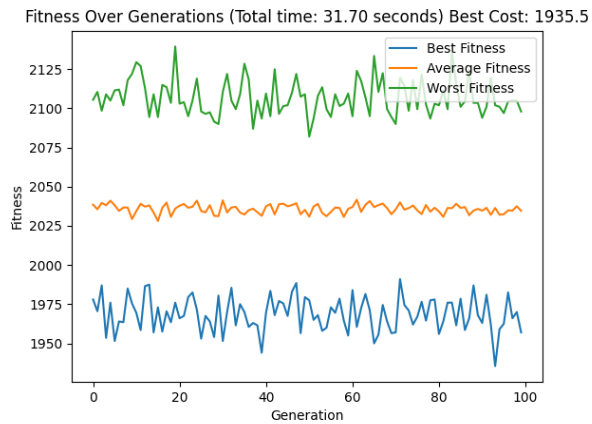

Figure 2: Showing Fitness values over generation for RA

EA Algorithm:

Statistic	Performance Metric	Average Time to Achieve
Mean	1892.85	NaN
Median	1896.00	0.595 seconds
Minimum	1872.00	0.597 seconds
Maximum	1932.00	0.617 seconds

Table 2 Showing Fitness Values for EA Over Generations

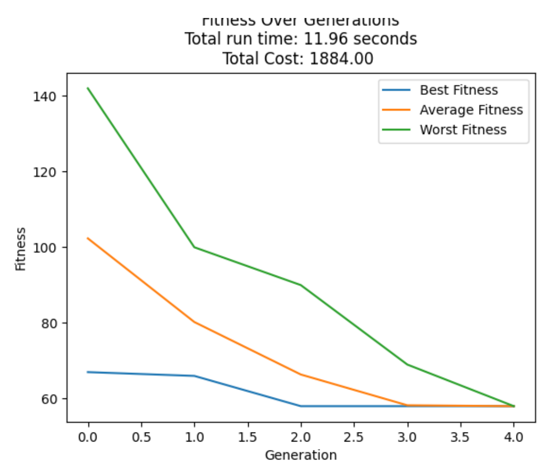

Figure 3 Showing Fitness Values for EA Over Generations

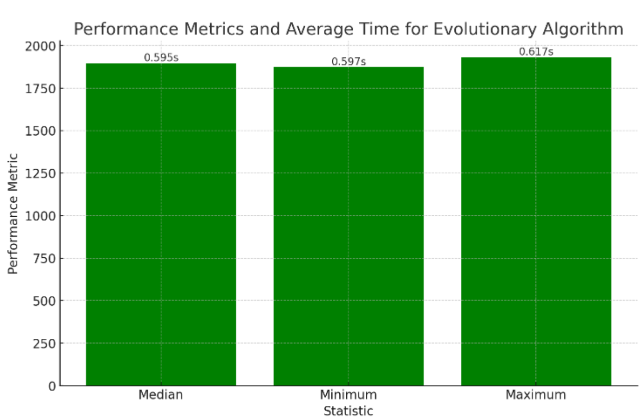

Figure 4 Block Graph showing Cost achieved in time for EA

Deductions:

- Given the very small p-value, we reject the null hypothesis.
- The Evolutionary Algorithm, particularly with a configuration of a smaller population size (10), lower mutation rate (0.05), and a moderate number of generations (100), outperforms the Random Algorithm in both cost and time efficiency when it comes to finding the best solution. If tuned perfectly
- The results suggest that a well-tuned EA can provide more cost-effective solutions in a shorter amount of time if tuned properly for the parameters compared to a random search approach.

Experiment 2:

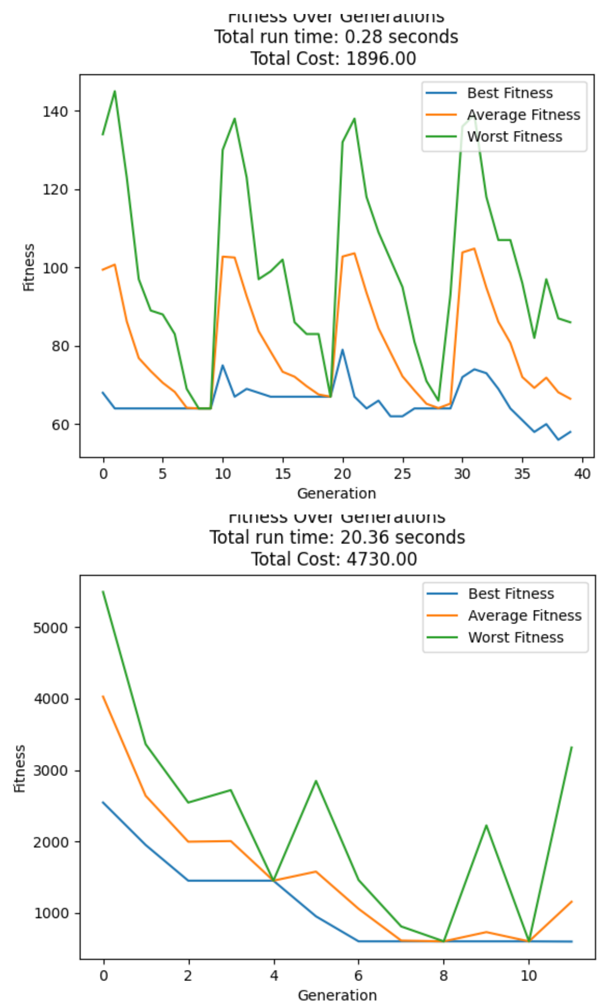

Figure 5 Showing the Effect Of Immigration Method on EA

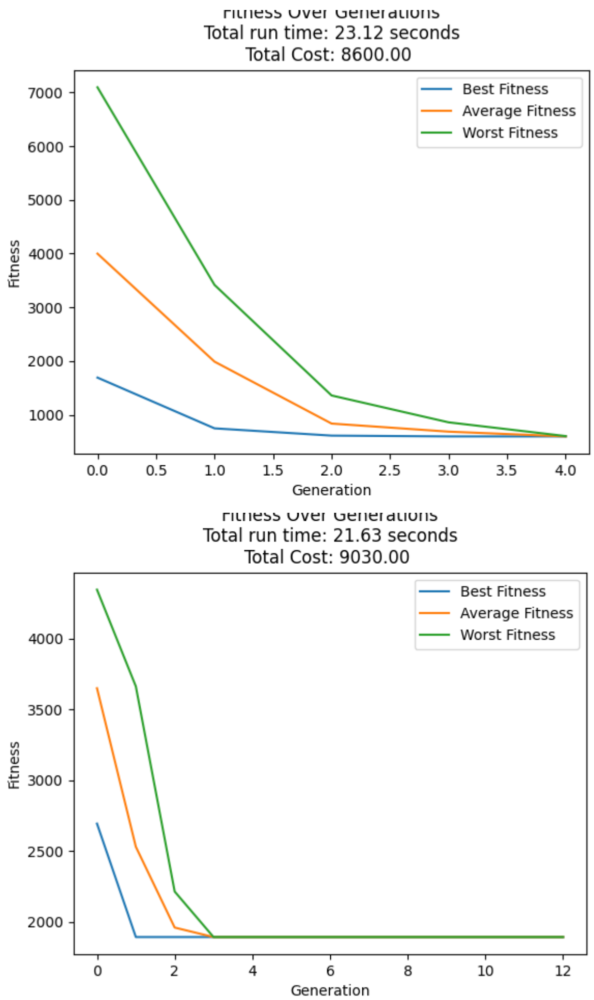

Figure 6 Showing the Effect of No Immigration Method on EA

Deductions:

- Immigration methods in the Evolutionary Algorithm significantly impact cost efficiency, with varying intensities and quantities leading to diverse cost outcomes.
- No clear correlation between immigration settings and computation time is observed, indicating their primary influence on solution quality rather than speed.
- Optimal tuning of immigration parameters is crucial for enhancing the cost-effectiveness of the EA in the cutting stock problem.

Experiment 3:

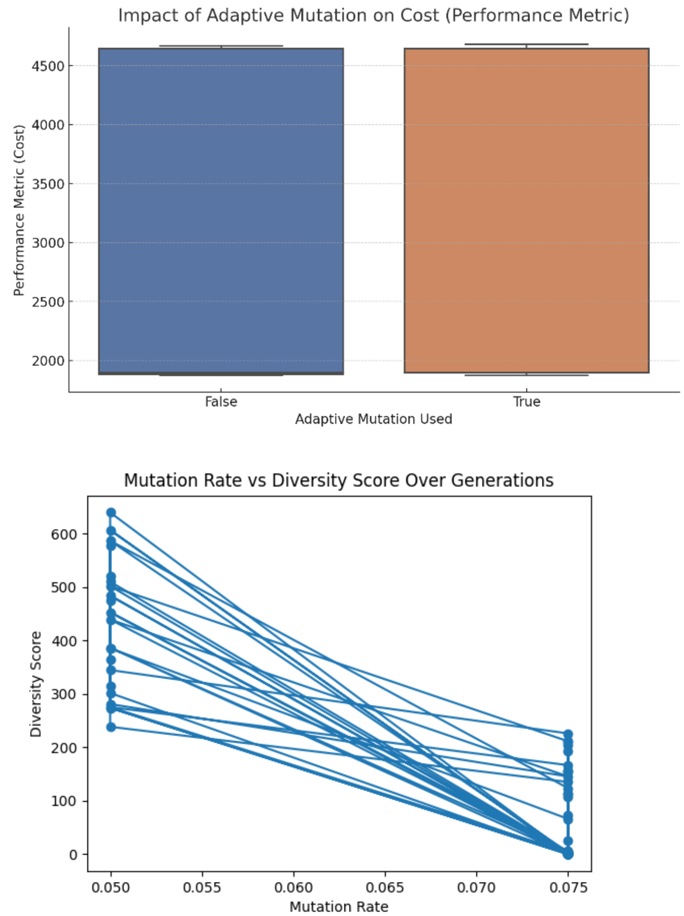

Figure 7: Test Results Conducted for Adaptive Mutation Rate

Deductions:

- Adaptive mutation in the EA doesn't affects solution costs that intensely but it does help the population escape local optima if it the mutation rate is intense, with varying effectiveness based on the mutation parameters.

Conclusion:

Key Insights from Evolutionary Algorithm Experiments:

1. **EA vs. Random Search**: EA outperforms random search in cost and time, particularly with optimal parameter tuning.
2. **Parameter Tuning:** Fine-tuning population size, mutation rate, and generations is crucial for EA efficiency.
3. **Immigration Method:** It significantly influence cost efficiency, the convergence of

the Algorithm. A better tuned immigration method can produce better results.

4. **Adaptive Mutation:** Doesn't influence the solution a lot unless it's a higher mutation rate. It helps the population escape local optima.

In my conclusion, I have been able to successfully verify immigration method does significantly affect the cost of the solution while adaptive mutation rate can lead to better solutions only if it is tuned perfectly. But it doesn't affect the efficiency significantly.

References:

- Liang, K.H., Yao, X., Newton, C. and Hoffman, D., 2002. A new evolutionary approach to cutting stock problems with and without contiguity. Computers & Operations Research, 29(12), pp.1641-1659.
- Harry Goldingay, Evolutionary Algorithms evol_alg.pdf
- Chiong, R. and Beng, O.K., 2007. A Comparison between Genetic Algorithms and Evolutionary Programming based on Cutting Stock Problem. Eng. Lett., 14(1), pp.72-77.
- Experiment Data Used for Deduction: https://drive.google.com/file/d/1Niz8oE6mBuJGNLb2Y8gkJQBYZEdgsBWf/view?usp=drive_link, https://drive.google.com/file/d/1UbG6uSCN-NRK9Rwa9aNYg3XsWADoFVSC/view?usp=drive_link, https://drive.google.com/file/d/1D1lc3ZARpqbWCha8h28FjJ1G2g8YyqIb/view?usp=drive_link
- Krzysztof Michalak,Evolutionary Algorithm Using Random Immigrants for the Multiobjective TravellingSalesman Problem,Procedia Computer Science,Volume 192,2021,Pages 1461-1470,ISSN 877-0509,https://doi.org/10.1016/j.procs.2021.08.150.(https://www.sciencedirect.com/science/article/pii/S1877050921016446)
- International Journal of Computer Applications (0975 – 8887) Volume 175– No. 10, August 2020 49 A Rank based Adaptive Mutation in Genetic Algorithm Avijit Basak IBM India Pvt. Ltd.
    - https://arxiv.org/ftp/arxiv/papers/2104/2104.08842.pdf
- A. E. Eiben, J. E. Smith: Introduction to Evolutionary Computing, Chapters 2, 3, 4, 5 (chapter 1 also for background)