Optimizing Flow-Shop Scheduling: Comparative Analysis of Heuristics for Minimizing Make-Span

INTRODUCTION

CE 2.1

I have done my bachelors in industrial engineering from [redacted] Also I took training for professional development in "Professional trends in Industrial Engineering" in [redacted] and [redacted]. In my academic course "Production planning and Control" which I pursued in my 6th semester; I did a project titled "Optimizing Flow-Shop Scheduling: Comparative Analysis of Heuristics for Minimizing Make-Span". The duration of the project was [redacted].

BACKGROUND

CE 2.2.1

This project focused on flow-shop scheduling, which involved arranging jobs in a specific order to minimize the make-span. I knew that a flow-shop consists of machines that process jobs in a predetermined sequence. The project focused on discussing the Permutation Flow-shop Scheduling. I considered factors such as limited resources, known job processing times, and no preemption. The objectives of the project included minimizing the make-span, providing a schedule for all jobs and machines, and comparing different heuristics to find the most suitable approach. The ultimate goals of the scheduling were to effectively utilize staff, equipment, and facilities, while minimizing client waiting time and inventories. The effectiveness of the proposed approaches is evaluated using Gantt charts.

CE 2.2.2

I proposed the following methodology for this project.

1. Literature Review: Constructive, Improvement Heuristics and Metaheuristics
2. Research Design: Problem Statement, Hypotheses
3. Data Collection: Job Processing Times, Machine Constraints.
4. Experimental Setup: Evaluation Metrics
5. Proposed Methodology: Palmer, Gupta, Johnson and Genetic Algorithms
6. Initialization of Population: Selection of Mechanism, Crossover Operators, Mutation Operators and Termination Criteria
7. Implementation: Tools, Development Process, Testing and Validation
8. Results and Analysis: Comparison of Scheduling Methods, Performance Evaluation of Genetic Algorithm and Statistical Analysis
9. Conclusion: Summary of Findings, Practical Implications and Recommendations

CE 2.2.3

I effectively communicated with my project advisor and collaborated with my team members to ensure smooth project execution. I actively engaged in discussions, sought guidance, and shared progress updates with the advisor and team. Additionally, I diligently followed established engineering practices and methodologies to ensure the project's adherence to quality standards and best practices in the field of industrial engineering. The hierarchy of this project was:

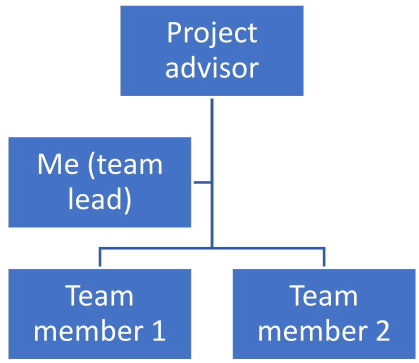

Figure 1: Hierarchy

PERSONAL ENINEERING ACTIVITY

CE 2.3.1

I knew that a flow-shop involves a unidirectional flow of work, where jobs follow a similar processing order through the machines. Production in a flow-shop requires arranging different activities to minimize the make-span. PFSP, introduced by Pinedo, focuses on maintaining the sequence of the jobs on the initial machine throughout flow-shop. This problem highlights the significance of logical sequencing and scheduling in the flow-shop environment. I was aware of the key considerations for this problem, including machine limitations, job availability, sequential arrangement, known processing times, and the restriction on preemption.

CE 2.3.2

It's important to note that different sequences based on processing time can lead to varying processing time. The goal was to find the sequence that minimized processing time. Scheduling, on the other hand, involved allocating constrained resources to tasks over time. It entailed assigning specific time slots for completing jobs, with the primary objective being to minimize processing time. The significance of this work lay in determining the order of job handling at work centers, resulting in an organized job list. It was particularly valuable when dealing with limited capacity and aimed to optimize resource utilization and minimize idle time at each machine. The objectives of my project included minimizing the make-span, creating schedules for jobs and machines, selecting suitable heuristics based on comparisons, and addressing the flow shop scheduling problem within a flexible manufacturing system. I aimed to achieve effective utilization of staff, equipment, and facilities, as well as reduce client waiting time and inventory levels.

CE 2.3.3

I conducted research and did literature review on Constructive Heuristics, Improvement Heuristics, Metaheuristics for a better understanding of the project. This was not only a necessary step for initializing the project but also a significant process by which I addressed my engineering practice. I created a comparative table of the literature review.

CE 2.3.4

In the next part I discussed the different algorithms, I understood the application of Palmer's and Gupta's heuristics for job scheduling in a flow shop environment. Using Palmer's heuristic, I calculated the optimal sequence by arranging jobs in non-increasing order of a specific slope. As an example, for a 10-job and 10-machine problem, the optimal sequence obtained was {3, 1, 10, 5, 6, 2, 8, 7, 9, 4}. I also implemented Gupta's heuristic, which involves forming groups of jobs based on their processing times on the first and last machines, and then finding the minimum of certain values to determine the optimal sequence. For a 4-machine and 7-job problem, the optimal sequence obtained using Gupta's heuristic was {G1, G2, G5, G7, G4, G3, G6}, resulting in a make-span of 85.

CE 2.3.5

I implemented the Johnson Algorithm for job sequencing in a machine scheduling problem. By identifying the job with the minimum time on the machines and placing it either at the beginning or end based on the machine it belongs to, I obtained the sequence {J1, J2, J3, J4} for a 4-job scheduling problem.

I also explored the Genetic Algorithm (GA) for optimization. In the initialization phase, I considered two techniques: random initialization and heuristic initialization. Fitness Proportionate Selection, specifically the Roulette Wheel Selection method, was used to select parents based on their fitness values.Crossover, an essential step in GA, involved combining genetic material from multiple parents to generate offspring. One Point Crossover and Multi Point Crossover were implemented to swap genetic segments. Mutation, used to introduce diversity, was achieved through Bit Flip Mutation and Swap Mutation. These operators randomly altered the chromosome to explore new solutions. To determine the termination condition for the GA, I considered options such as reaching a specific fitness value, a maximum number of generations, or no change in the population for a certain number of iterations.

CE 2.3.6

To assess the performance of the methods, I calculated the percentage increase over the optimum solution using the formula: % increase over optimum = (Heuristic solution - optimum solution) / (optimum solution) * 100

I implemented the Johnson, Palmer, Gupta heuristics, and Genetic Algorithm (GA) using Java programming. The problem instances covered a range of 2 to 100 jobs and 2 to 20 machines. In the case of the Genetic Algorithm (GA), I employed single point crossover and double point crossover, combined with shift mutation. Parameters of GA were: Probability of the crossover was 0.6, and of mutation was 0.1, and a total of 1000 generations. The outcomes obtained by applying these approaches to different problem sizes are provided. Among them, the Johnson algorithm demonstrated superior performance in the majority of cases, followed by Gupta's heuristic and the GA. The Palmer heuristic had the highest average increase over the optimum solution.

Problem	Johnson	Palmer	Gupta	GA
2*2	2	3.5	3.5	2
5*2	-	4.2	4.5	4.5
10*5	-	6.1	6.45	6.1
20*5	-	10.68	12.3	4.46
25*10	-	15.37	23.28	7.84
50*5	-	5.25	12.5	1.34
50*10	-	14.10	22.10	8.50
50*20	-	16	24.66	9.59
Average		9.4	13.66	5.541

Table 3: Results of various sizes of Jobs and Machines for Different Heuristics and GA

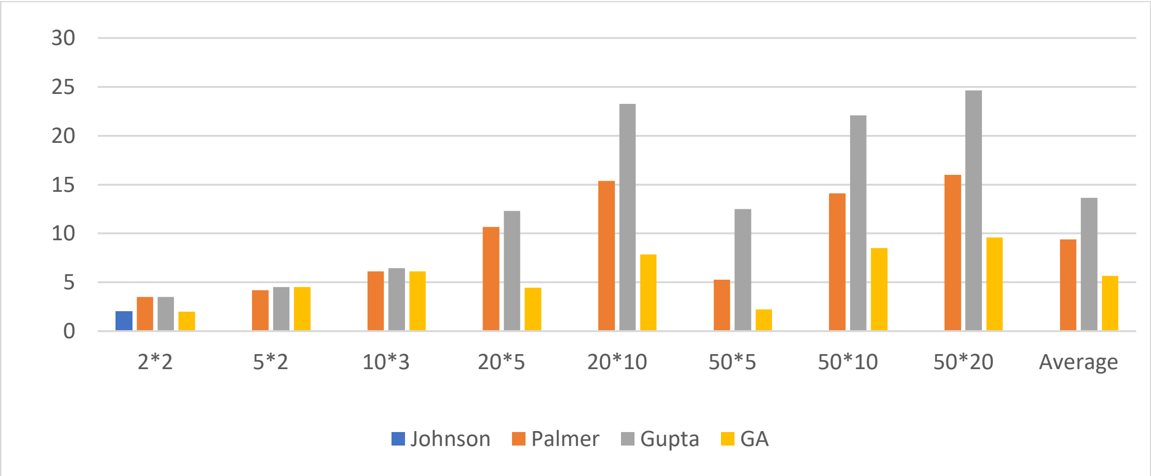

Figure 2: Bar chart of Increase over optimum in percentage average

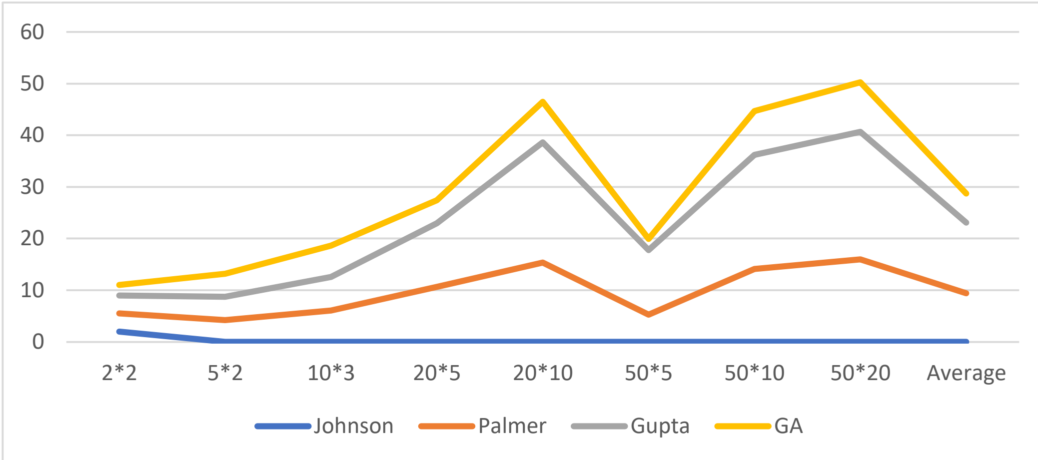

Figure 3: Trend of Increase over optimum in percentage average

S. No.	**N*M**	**Single Point Crossover** **Shift Mutation**	**Two Point Crossover** **Shift Mutation**
1	5*4	**208**	208
2	10*5	325	324
3	20*5	405	404
4	20*10	487	486
5	50*5	1022	1022
6	30*15	1275	1205
7	50*20	1701	1666
8	80*10	1842	1820
9	80*20	2308	2306
10	100*10	2210	2206

Table 4: Results for GA for Single and Double Point Crossover

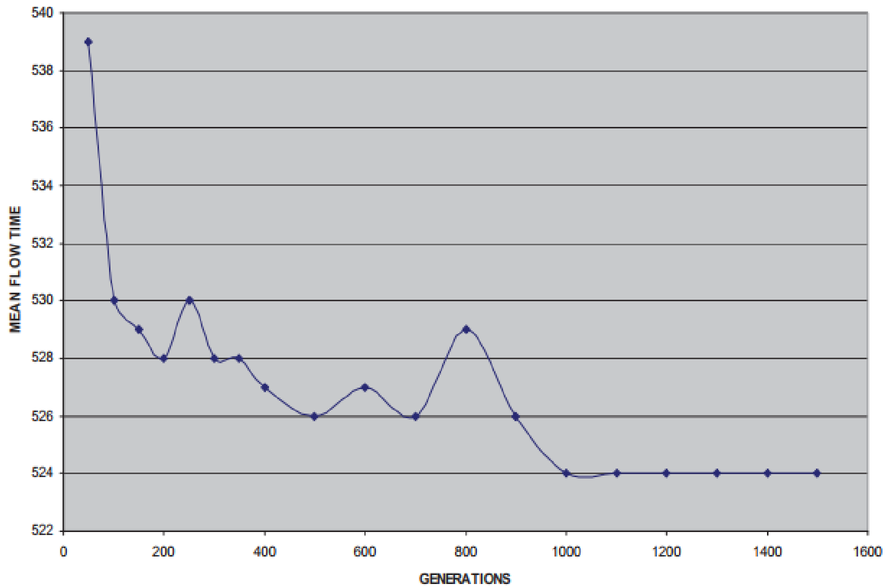

Figure 4: Mean Flow Time Vs Generations in GA

CE 2.3.7

I acknowledged the effectiveness of the Genetic Algorithm (GA) in addressing the flow shop scheduling problem. I conducted evaluations using various problem instances of different sizes to assess the GA's performance. To achieve this, I utilized single point crossover and double point crossover done with the shift mutation shift

The findings indicated that for larger problem sizes, the double-point crossover produced superior results in terms of the make-span. Recognizing the potential for further improvement, I explored alternative crossover and mutation operators within the GA. The mutation operation involved exchanging partial sequences (chromosomes) to determine the optimal sequencing order and reach optimal solutions.

Notably, Gupta's heuristic demonstrated an increase of over 13% compared to the optimum solution, despite claims made by the author regarding its superiority over Palmer's algorithm, which served as a benchmark in my study. In contrast, Palmer's heuristic consistently outperformed Gupta's heuristic across all problem sizes, with statistical analysis confirming significant differences between the two methods.

CE 2.3.8

I discovered that the Genetic Algorithm (GA) emerged as the most effective heuristic among the tested methods. It consistently achieved superior performance, surpassing other approaches by a significant margin. The GA exhibited robustness, maintaining consistent results regardless of the number of jobs. In contrast, Johnson's, Gupta's, and Palmer's heuristics proved to be sensitive to variations in job count. A similar pattern was observed with the number of machines, where the best outcomes were obtained with five machines for different methods. Additionally, it is noteworthy to acknowledge the correlation between the variables denoting the jobs (n)/machines (m) number, as demonstrated in the investigation. I identified the instances where jobs and machines were 50 and 20, as well as those where jobs and machines were 100 and 20 respectively. These specific instances pose ongoing difficulties, with no established optimal solution identified thus far.

SUMMARY

CE 2.4.1

I extensively researched and analyzed different methodologies and algorithms to tackle the flow-shop scheduling problem and minimize the make-span. Through a comprehensive literature review, I gained insights into constructive heuristics, improvement heuristics, and metaheuristics. I compared and summarized these heuristics in a table to understand their applicability. During my investigation, I specifically focused on Palmer's and Gupta's heuristics, and I was able to identify optimal sequences by applying these methods. Furthermore, I explored the Genetic Algorithm (GA) as an optimization approach. I implemented various techniques such as random and heuristic initialization, fitness proportionate selection, and crossover (single-point and double-point), along with mutation (bit flip and swap). I carefully considered the termination criteria for the GA. Through my experimentation, I obtained results for different problem sizes, providing valuable insights into the effectiveness of these approaches. Based on my findings, it was observed that the Genetic Algorithm (GA) consistently outperformed other heuristics, demonstrating minimal increases in make-span. Through my extensive research, I uncovered that the utilization of two-point crossover done with the shift mutation exhibited enhanced effectiveness when dealing with larger-scale problem instances. Moreover, I realized the possibility of enhancing the GA even further by investigating alternative crossover and mutation operators.

CE 2.4.2

Interestingly, I found out that Gupta's heuristic resulted in an increase of more than 13% compared to the optimum solution, which contradicted claims of its superiority over Palmer's algorithm. On the other hand, I consistently observed that Palmer's heuristic outperformed Gupta's heuristic in all problem sizes.

Through analysis, I emphasized the robustness of the GA as it consistently performed optimal regardless of the job numbers, while the other heuristics showed sensitivity to the job count. I also discovered that the best outcomes for various methods were achieved with five machines. Additionally, I discussed the variables representing the jobs numbers and the machines, and I identified the instances where jobs and machines were 50 and 20, as well as those where jobs and machines were 100 and 20 respectively, as the most challenging ones without established optimal solutions.

Overall, my project on flow-shop scheduling provides valuable insights into the application of different heuristics and the effectiveness of the Genetic Algorithm. These findings contribute to the field of industrial engineering and suggest potential areas for further research and improvement in optimizing flow-shop scheduling problems.