Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/BA08.20250186
Published: December 20, 2025

Optimizing Garment Assembly Line Efficiency Using Ranked Positional Weight and Evolutionary Algorithms: A Comparative Analysis

Kazi Fahim, Sharafat Hossain and Joyanta Debnath
Industrial and Production Engineering Department
National Institute of Textile Engineering and Research
Dhaka, Bangladesh
kazifahim.ipe@gmail.com, sharafat.ipe@gmail.com, joyantadeb88@gmail.com

Nurul Ahad Choudhury
School of Industrial Engineering and Management
Oklahoma State University, Stillwater, OK 74078, USA
nachoudhury@niter.edu.bd

Abstract

The purpose of this study is to develop an effective method for balancing garment assembly line to improve efficiency, reduce bottlenecks, and minimize non-value-added time. The research addresses limitations in manual task allocation by using Ranked Positional Weight (RPW), evolutionary optimization techniques for dynamic workstation planning, and simulation. A jacket sewing line consisting of 27 operations was selected for analysis using RPW and allocated using a custom Python algorithm under Basic Pitch Time (BPT) constraints. Standard Minute Values were measured through time studies, and precedence relationships were constructed. Two evolutionary optimization methods – Genetic Algorithm (GA) and Differential Evolution (DE) were applied to increase workload balancing, and the proposed layout was validated using simulation. DE achieved a more compact layout with 16 workstations and higher line efficiency (77%) compared to GA's 18 workstations and efficiency (71%). GA resulted in slightly higher total output (509 units) over DE (488 units) in 10 hours and maintained full compliance with precedence constraints, while DE had faster convergence but introduced minor violations. DE also had lower SMV variance, indicating better workload distribution. The proposed method provides an integrated method that was validated through simulation and offers a practical approach for balancing and optimizing line efficiency.

Keywords: Garment industry, Line Balancing, Ranked Positional Weight (RPW), Genetic Algorithm (GA), Differential Evolution (DE).

1. Introduction

The garment industry plays a crucial role in the global economy, particularly in countries like Bangladesh, where it is one of the largest sources of employment and export earnings. However, improving productivity and operational efficiency is essential to remain competitive in this industry (T. Islam & Halim, 2022). The garment industry in Bangladesh faces significant productivity challenges, particularly related to sub-optimized line balancing and bottlenecks. Inefficient task distribution along production lines often leads to bottlenecks, and certain workstations become overloaded while others remain underutilized, reducing overall efficiency (Akhter et al., 2019; Chowdhury et al., 2014). These bottlenecks slow production rates and increase non-value-adding time, contributing to higher operational costs in the supply chain of all types of industries (Ahamed et al., 2024; Hosen, 2020). Moreover, Arefin et al. (2022) highlighted that the lack of advanced planning tools and the reliance on manual processes further worsen these issues and make it difficult for industries to maintain consistent production flow and meet increasing demand.

One of the most effective strategies for optimizing production in garment manufacturing is line balancing, which ensures an even distribution of workload across all workstations (Hazır & Dolgui, 2019). Effective line balancing reduces bottlenecks and improves overall efficiency in an assembly line (Boysen et al., 2022; Dolgui & Proth, 2013). Therefore, addressing these bottlenecks through improved line balancing is crucial for enhancing productivity in the garment industry. Despite numerous studies using line balancing in enhancing garment industry productivity, they mostly rely on static or heuristic methods without using optimization-based techniques (Katiraee et al., 2023; Ponnambalam et al., 1999; Rahman et al., 2024). Additionally, very few studies have combined these algorithms with simulation to validate real-world applicability.

1.1 Objectives

The objective of this study is to develop an automated framework for garment layout planning that optimizes workstation allocation using RPW values and Basic Pitch Time (BPT) to minimize non-value-added time. Additionally, the framework aims to adapt Genetic Algorithm (GA) and Differential Evolution (DE) to achieve adaptive and efficient assembly line balancing to varying production conditions and achieve automatic assembly line balancing through Python-based data analysis and simulation techniques.

2. Literature Review

The Simple Assembly Line Balancing Problem (SALBP) was first used in research by Salveson (1955), which is a method used to allocate tasks evenly among workstations in an assembly line to ensure that no single station becomes a bottleneck (Ghosh & and Gagnon, 1989). Apparel industries face intense global competition marked by short product life cycles, high labor costs, shifting customer demands, growing consumer awareness, and the need for rapid, yet unpredictable, production timelines (Noor et al., 2022). Previous studies have shown that production efficiency can be significantly improved by task redistribution and line balancing (Ahmed et al., 2020; Schoenfelder et al., 2025). Bottlenecks in garment manufacturing lines often result in inefficiencies that affect production rates and labor productivity (Battaïa & Dolgui, 2013; Chen et al., 2012). Bappy et al. (2019) highlighted the importance of identifying and eliminating bottlenecks through work-sharing methods and task redistribution. Past research has also explored various approaches to address bottlenecks in garment production

Elahi et al. (2020) examined the impact of bottlenecks on production in the apparel industry and proposed methods such as lean techniques and line balancing to eliminate inefficiencies (Elahi et al., 2020). These methods help streamline operations and increase the productivity of garment assembly lines by minimizing non-value-added activities. This study builds on the existing body of research by applying line balancing techniques to a specific garment manufacturing context. Moreover, Teshome et al. (2024) emphasized that bottlenecks in garment production lines, especially in sewing operations, lead to inefficiencies such as idle time and longer cycle times, reducing overall productivity. Chen et al. (2012) similarly noted that balancing assembly lines in the garment industry is a complex task, as uneven task distribution across workstations can result in waiting times and increased production costs. These findings indicate the need for more sophisticated line balancing techniques and better knowledge dissemination to improve productivity and reduce bottlenecks in the industry.

3. Methods

This research applies line balancing methods to optimize the process layout of a garment assembly line and aims for efficient task distribution across workstations. The research methodology involves operational breakdown, Ranked Positional Weight (RPW) calculation, precedence graph construction, task prioritization, and validation to ensure a balanced and practical layout.

3.1 Ranked Positional Weight (RPW)

RPW prioritizes tasks based on their cumulative importance to the assembly process. For each task $\mathbf{P_i}$, the RPW is calculated as the task's own operating time plus the sum of the operating times of all its successor tasks (Achmadi et al., 2023). In mathematical notation, this is expressed as:

$$RPW_i = t_i + \sum t_k \quad (1)$$

Where $\boldsymbol{t_i}$ is the Operational time (SMV) for task i and k denotes the set of successor tasks for task i. This equation ensures that tasks with longer downstream processing chains receive higher priority. Operational Data with RPW value are shown in Table 1. To automate workstation allocation, a custom Python script was developed. The algorithm uses RPW values, SMVs, and precedence constraints to assign tasks within a fixed cycle time (1.20 minutes per workstation). Machine availability is also validated to ensure the required machine for each task. The implementation was done in Python 3.9 using the **panda's** library. The process is as follows:

$$\text{Sorted Tasks} = Sort(P_i, \text{key} = \text{RPW}(i), \text{order} = \text{descending}) \quad (2)$$

$$\sum_{\text{Ws}=1}^{\text{n}} \text{Ws} \cdot \text{x}(\text{Ws}, \text{i}) \geq \sum_{\text{Ws}=1}^{\text{n}} \text{Ws} \cdot \text{x}(\text{Ws}, \text{j}) \quad \text{, all i and j} \in \text{P(i)} \qquad (3)$$

$$\text{P}_\text{i} \in \text{Ws(n), then Req}(\text{P}_\text{i}) \subseteq \text{M/c(n)Types of Machines available} \qquad (4)$$

$$\sum_{\text{i}\in\text{Ws(n)}} \text{t}_\text{i} \leq \text{BPT}. \qquad (5)$$

3.2 Optimization methodology

Differential Evolution (DE) and Genetic Algorithm (GA) methods were used for optimizing workstation allocation. These evolutionary algorithms aim to minimize SMV variance, enhance line efficiency, and satisfy process constraints.

3.2.1 Genetic Algorithm (GA) methodology:

Genetic algorithm (GA) is an optimization process based on a population-based search algorithm, which utilizes the concept of survival of the fittest (Katoch et al., 2021).There was a maximum of 200 Generations and each generation consists 100 population. The adopted GA process in this study followed the following steps.

Step 1: Encoding and Initialization. Each chromosome represents workstation allocations, e.g., [1, 2, 1, 3...].

Step 2: Fitness Evaluation. This function evaluates the quality of a chromosome *S* representing a complete workstation layout, seeking to minimize the imbalance in workload across workstations and penalizes constraint violations.

$$f(S) = Penalty_{Ws} + Penalty_{pred} + Penalty_{cy} \qquad (6)$$

Where, $\text{Penalty}_{\text{ws}}$ is defined as workstation Penalty. This penalty will prevent the generation of a large number of unique workstations. Penalty will be counted as (Number of Workstation)^2.

$$Penalty_{Ws} = (Number\ of\ Workstation)^2 \quad (7)$$

$\text{Penalty}_{\text{cy}}$ is the cycle time Penalty. If the total cycle time of a workstation exceeds BPT, then

$$\sum_{Ws=1}^{n} \max(0, total\ Cycle\ time\ of\ Ws - BPT)^2 \qquad (8)$$

Step 3: Genetic Operators selection. This method includes U tournament or roulette-wheel selection to choose parents. This operator selects the better solution (lower fitness value) between two randomly chosen chromosomes.

$$S = \begin{cases} S_{r1} & if\ f(S_{r2}) \geq f(S_{r1}) \\ S_{r2} & \text{otherwise} \end{cases} \qquad (9)$$

1. **Crossover:** Our method focused on performing one-point crossover.

$$Child_1 = [S_{r1}[1 : cut_1], S_{r2}[cut_1 + 1 : N]] \qquad (10)$$

$$Child_2 = [S_{r2}[1 : cut_1], S_{r1}[cut_1 + 1 : N]] \qquad (11)$$

Where, N indicates the number of total tasks.

2. **Mutation:** Mutation is used to introduce random changes to genes.

$$S_{Mutant} = \begin{cases} random\ workstation \neq s_{k,r} & \text{with probability of 0.2} \\ s_{k,r} & \text{otherwise} \end{cases} \quad (12)$$

The mutation rate is 0.2, which prevents all chromosomes from becoming too alike.

3.2.2 Differential Evolution (DE) methodology

To solve the assembly line balancing problem with precedence and cycle time constraints, a DE algorithm was implemented in Python. The purpose was to distribute Standard Minute Values (SMV) across a feasible number of workstations such that the line is balanced while minimizing constraint violations (S. M. Islam et al., 2012).

Step 1: Initialization. At generation G, the number of separate solutions was generated, a population of N solutions $\mathrm{S_r} = [\mathrm{s_{r1}}, \mathrm{S_{r2}}, \mathrm{S_{r1}} \ldots\ldots, \mathrm{S_{rn}}]$ where n indicates the number of tasks. Each individual $\mathrm{S_{rk}}$ indicates the workstation allocation for process k. Initial individuals for each solution were randomly generated.

Step 2: Mutation. The next goal was to generate a mutant vector. Mutation was done to generate a diverse vector compared to the best solution from N populations of the current generation.

$$\mathrm{MV_{r,G+1}} = \mathrm{S_{Best,G}} + \mathrm{Factor} \cdot \{(\mathrm{s_{1,G}} - \mathrm{S_{2,G}}) + (\mathrm{s_{3,G}} - \mathrm{S_{4,G}})\} \quad (13)$$

Where, $\mathrm{S_{Best,G}}$ = Best individual solution in the current generation, "Factor" refers to the scaling factor. A high value (e.g. 0.8) means a bigger jump in the exploration. Our current Factor value is 0.1, and $\mathrm{S_{1,G}}$, $\mathrm{S_{2,G}}$, $\mathrm{S_{3,G}}$, $\mathrm{S_{4,G}}$ are randomly selected from the current generation G.

Step 3: Crossover. A binomial crossover mechanism was used to diversify the mutation vector and the current solution, ensuring that at least one gene comes from the mutant. From the crossover, a trail vector $\mathrm{T_{k,r,G+1}}$, was generated. A fixed crossover probability was defined and a random number belonging to [0,1] was generated for each random gene, defined as random,

$$T_{k,r,G+1} = \begin{cases} MV_{k,r,G+1} & \text{If } \ random_k \leq Crossover\ Rate \ \ or\ k = k_{random} \\ S_{k,r,G} & \text{Otherwise} \end{cases} \quad (14)$$

Where, $\mathrm{k_{random}}$ is an index that is selected at random and $\mathrm{S_{k,r,g}}$ defines the parent solution of Generation G. Crossover rate was set at 0.8.

Step 4: Selection. This selection operator keeps the better solution between the trial vector $\mathrm{T_{r,G+1}}$ and the current vector $\mathrm{S_{r.G}}$ (Rahnamayan et al., 2008), to ensure that the DE algorithm only propagates improved or equally performing layouts in garment line balancing.

$$\mathrm{S_{r,G+1}} = \begin{cases} \mathrm{T_{r,G+1}} & \text{If } f(\mathrm{S_{r,G}}) \geq f(\mathrm{T_{r,G+1}}) \\ \mathrm{S_{r,G}} & \text{otherwise} \end{cases} \quad (15)$$

Step 4.1: Fitness Function. The fitness function penalizes cycle time violations, precedence constraint violations, excessive workstation usage, and unassigned processes. It is defined as:

$$\mathrm{f(S)} = \mathrm{Penalty_{ws}} + \mathrm{Penalty_{pred}} + \mathrm{Penalty_{cy}} \quad (16)$$

4. Data Collection and Preprocessing

The preprocessing stage organizes and validates task parameters to enable accurate task prioritization and workstation generation. Operation breakdown analysis is conducted for a jacket production line, entailing 27 specific operations. Standard Minute Value (SMV) is measured using an analog stopwatch for each operation. Each operation was measured by a time study process under standard conditions. Experienced operators performed 5–10 cycles per task. Basic times (in seconds) were averaged and adjusted to a 100% performance rating, and multiplied by a 15% allowance for personal allowance, fatigue allowance, and contingency factors. SMVs were converted to minutes (rounded to two decimals) and tabulated by the following formula,

$$SMV = Average\ observed\ cycle\ time \times Rating \times (1 + Allowance) \quad (17)$$

For this considered style customer demand is 500 pieces in a day. Available time per day is 10 hours. Then

$$\text{TAKT Time} = \frac{\text{Total Available time per day(min)}}{\text{Customer Demand}} = \frac{600}{500} = 1.20\ \text{min/pieces} \qquad (18)$$

The Basic Pitch Time (BPT) of 1.20 min is directly derived from the TAKT time to synchronize the line output with customer demand.

Table 1 shows the operation breakdown of a Jacket, precedence "Pre" refers to the immediately prior task, with "0" indicating no in-line predecessor (externally supplied component). This sequence establishes dependencies between operations to maintain production flow. The Starting process is "Facing joint at hand pocket lower bag position." Process 1.01, and the final process is Top Stitch at the cuff joint position, Process 5.02. Total SMV is 15.10 Min. The machine "LST" refers to the "Lock Stitch Machine", "OL-5" refers to the "Overlock five thread machines", "BRTK" refers to the "Bartack Machine "and "OL-3" refers to the "Overlock three thread machines".

Table 1. Operation breakdown with predecessor and RPW Value

Process Code	**Process Name**	**Machine Name**	**Process SMV**	**Pre task**	**RPW Value**
1.01	Facing joint at hand pocket lower bag position	OL-5	0.39	0	13.07
1.02	Top stitch at hand pocket lower bag facing joint position	LST	0.37	1.01	12.68
1.03	Zipper joint at hand pocket mouth position	LST	0.54	1.02, 0	12.31
1.04	Upper bag joint at hand pocket mouth position	LST	0.51	1.03	11.77
1.05	Top stitch at hand pocket mouth upper position	LST	0.60	1.04	11.26
1.06	Lower bag joint at hand pocket mouth position	LST	0.62	1.05	10.66
1.07	Hand pocket bag close upper position	OL-5	0.44	1.06	10.04
1.08	Top stitch at hand pocket mouth lower	LST	0.44	2.01	9.1
1.09	Bartack at hand pocket mouth position	BRTK	0.36	1.08	8.66
2.01	Overlock at side part joint front part position	OL-3	0.50	1.07, 0	9.6
3.01	Side part joint at back part position	OL-5	0.55	6.03, 0	7.22
4.01	Sleeve side seam upper position	OL-5	0.63	0	8.1
5.01	Overlock at cuff joint position	OL-3	0.43	8.03	1.23
5.02	Top stitch at cuff joint position	LST	0.80	5.01	0.8
5.03	Cuff joint at sleeve lower position	LST	0.80	4.01	7.47
6.01	Collar inside center front part joint	LST	0.40	9.03	3.06
6.02	Collar inside center front part close neck position	LST	0.32	6.01	2.66
6.03	Collar join at neck position through shoulder	LST	1.08	1.09, 0	8.3
6.04	Bartack at neck position	BRTK	0.31	6.02	2.34
7.01	Overlock at inner plkt edge position	OL-3	0.23	9.01, 9.02	6.07
7.02	Zipper joint at inner plkt position	LST	0.45	7.01, 0	5.84
8.01	Mark at center front zipper joint position	Hand Work	0.28	7.02	5.39
8.02	Zipper joint at center front position with mark	LST	1.00	8.01	5.11
8.03	Top stitch at center front Zipper joint position	LST	0.80	6.04	2.03

Process Code	**Process Name**	**Machine Name**	**Process SMV**	**Pre task**	**RPW Value**
9.01	sleeve joint at armhole position in left side	OL-5	0.60	3.01, 5.03	6.67
9.02	sleeve joint at armhole position in right side	OL-5	0.60	3.01, 5.03	6.67
9.03	Binding at bottom hem position	LST	1.05	8.02	4.11

5. Results

5.1 Initial Allocation of Ranked Positional Weight (RPW) Algorithm

Using the complete dataset, tasks were grouped based on RPW values and standard minute values (SMV), with a focus on balancing workloads close to 1.20 SMV per workstation. Machine constraints and logical workflow progression were also considered. Processes are grouped based on RPW and assigned with cycle time constraints in Python. The initial allocation is presented in Table 2.

Table 2. Initial workstation allocation

Workstation	Total SMV (min)	Process Index	Machine Type(s)
1	0.76	1.01, 1.02	OL-3, LST
2	1.05	1.03, 1.04	LST
3	0.60	1.05	LST
4	1.06	1.06, 1.07	LST, OL-5
5	0.94	2.01, 1.08	OL-3, LST
6	0.36	1.09	BRTK
7	1.08	6.03	LST
8	0.63	4.01	OL-5
9	0.80	5.03	LST
10	1.15	3.01, 9.01	OL-5
11	0.83	9.02, 7.01	OL-5, OL-3
12	0.73	7.02, 8.01	LST, Hand Work
13	1.00	8.02	LST
14	1.05	9.03	LST-F
15	1.03	6.01, 6.02, 6.04	LST, BRTK
16	0.80	8.03	LST
17	0.43	5.01	OL-3
18	0.80	5.02	LST

5.2 GA and DE Algorithm

The proposed method validates that each workstation complies with the Basic Pitch Time (BPT), maintaining cycle time limits. Tasks are assigned in proper sequence while adhering to all precedence constraints. Additionally, the workload is distributed evenly across workstations by minimizing imbalance and enhancing line efficiency. The work allocation of GA and DE are shown in Table 3 and Figure 1.

Table 3. Optimized workstation allocations in GA and DE

Workstation	**GA – Total SMV**	**GA – Process Index**	**DE – Total SMV**	**DE – Process Index**
1	0.76	1.01, 1.02	1.16	1.01, 1.02, 6.01
2	1.05	1.03, 1.04	1.05	1.03, 1.04
3	0.6	1.05	1.15	1.05, 3.01
4	1.06	1.06, 1.07	0.62	1.06
5	0.94	2.01, 1.08	0.87	1.07, 5.01
6	0.36	1.09	0.95	2.01, 7.02
7	1.08	6.03	0.8	1.08, 1.09
8	0.63	4.01	1.08	6.03
9	0.8	5.03	0.63	4.01
10	1.15	3.01, 9.01	0.8	5.03
11	0.83	9.02, 7.01	1.2	9.01, 9.02
12	0.73	7.02, 8.01	1.14	7.01, 8.01, 6.02, 6.04
13	1	8.02	1	8.02
14	1.05	9.03	1.05	9.03
15	1.03	6.01, 6.02, 6.04	0.8	8.03
16	0.8	8.03	0.8	5.02
17	0.43	5.01	—	—
18	0.8	5.02	—	—

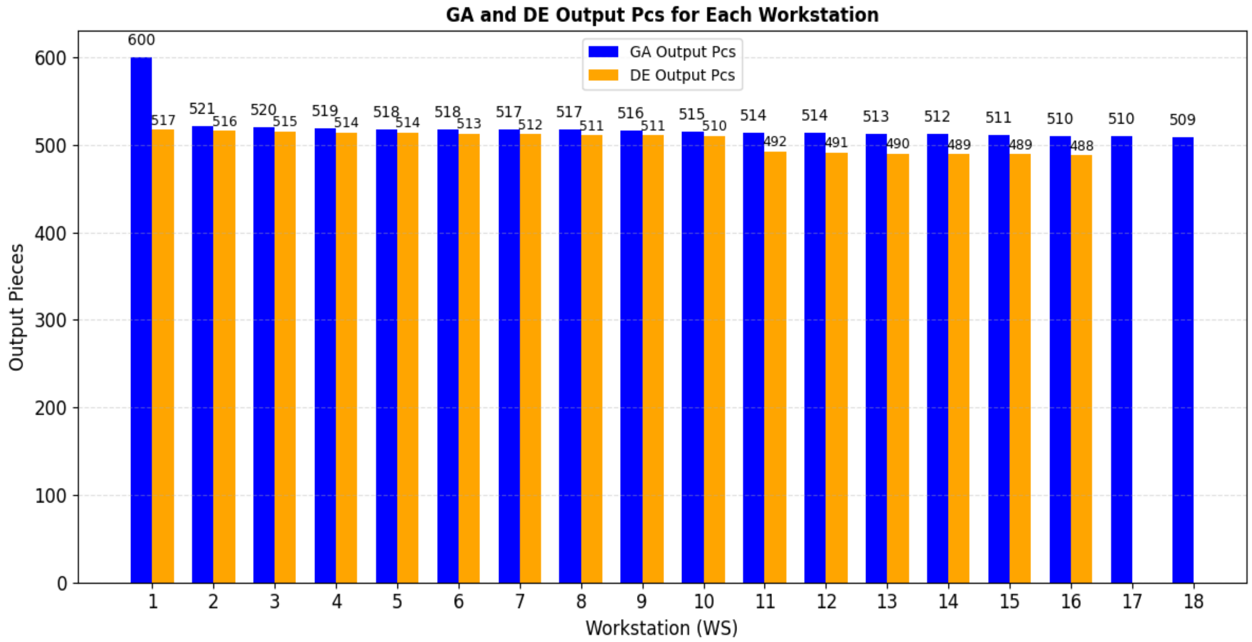

Figure 1. GA and DE output of pieces against workstation

To comprehensively assess the performance of the DE and GA optimization methods, the SMV values were utilized. SMV variance measures the uniformity of workload distribution across workstations. Lower variance implies better balancing and ultimately reduces idle times and improves overall efficiency, as shown in Table 4.

Table 4. Comparison of evaluation metrics between GA and DE

Metric	**GA Layout**	**DE Layout**
Number of Workstations	18	16
Total SMV (min)	15.1	15.1
Mean SMV (min)	0.8389	0.9438
SMV Variance	0.0492	0.0331
Interpretation	Less balanced	More balanced

Simulation Result:

Using eighteen workstations in Arena Simulation Software, the genetic algorithm (GA) simulation produced generally balanced line performance with a few important inefficiencies. Workstation 2 (WS2) displayed 99.9% usage, an average queue length of 39.03, and the highest accumulated wait times of 20,319 minutes. Therefore, indicating a main congestion point in the early stage of the line. Most later stations ran without notable queuing or delays, and stations including WS7, WS13, WS16, and WS18 reflected smooth flow behavior. A few stations (WS3, WS6, WS8, and WS17) showed underutilization, implying that work distribution might be improved still further. Overall, GA used 23 machines and generated a higher total output of 509 clothing in 10 hours simulation time length, and obtained a line efficiency of 71%. Total Number of input is fixed to 600 units. The GA simulation result is shown in Table 5.

Table 5. GA simulation result

Work station	**VA Time (min)**	**Total Time (min)**	**Utilization (%)**	**Queue Length Avg/Max**	**Number In**	**Number Out**	**Wait Time (min)**	**Notes**
WS1	0.76	0.76	76	0 / 0	601	600	0.00	Moderate utilization
WS2	1.15	40.15	99.9	39.03 / 79	600	521	20319	Severe bottleneck
WS3	0.60	0.60	52.0	0 / 0	521	520	0.00	Underutilized
WS4	1.06	1.06	91.8	0 / 0	520	519	0.00	High utilization
WS5	0.90	0.90	77.8	0 / 0	519	518	0.00	Moderate utilization
WS6	0.36	0.36	31.1	0 / 0	518	518	0.00	Low utilization
WS7	1.08	1.08	93.2	0 / 0	518	517	0.00	High utilization
WS8	0.63	0.63	54.3	0 / 0	517	517	0.00	Underutilized
WS9	0.80	0.80	68.8	0 / 0	517	516	0.00	Moderate load
WS10	1.15	1.15	98.8	0.18 / 1	516	515	0.11	Near-full capacity
WS11	0.83	0.83	71.2	0 / 0	515	514	0.00	Balanced activity
WS12	0.73	0.73	62.5	0 / 0	514	514	0.00	Moderate throughput
WS13	1.00	1.00	85.5	0 / 0	514	513	0.00	High throughput
WS14	1.05	1.05	89.6	0 / 0	513	512	0.00	High utilization
WS15	1.03	1.03	87.8	0 / 0	512	511	0.00	High utilization
WS16	0.80	0.80	68.1	0 / 0	511	510	0.00	Balanced load
WS17	0.43	0.43	36.6	0 / 0	510	510	0.00	Low utilization
WS18	0.80	0.80	67.9	0 / 0	510	509	0.00	Balanced throughput

The discrete event (DE) simulation involved 16 workstations evaluated under genetic algorithm-optimized line balancing. Workstation 1 (W1) demonstrated a severe bottleneck, reaching full utilization (100%) and accumulating the longest queue (max = 83) with a total wait time of 21,341 minutes. A secondary bottleneck appeared at W11, with near-full utilization (98.48%) and moderate queue buildup. In contrast, stations such as W4 and W9 showed underutilization, reflecting opportunities for workload redistribution. These results are instrumental in assessing flow efficiency and identifying constraints in the proposed production layout. Total Number of input is fixed to 600 units. The DE simulation result is shown in Table 6.

Table 6. DE simulation results

Work station	**VA Time (min)**	**Total Time (min)**	**Utilization (%)**	**Queue Length Avg/Max**	**Number In**	**Number Out**	**Wait Time (min)**	**Notes**
WS1	1.16	42.44	100	41.38 / 83	601	517	21,341	Severe bottleneck
WS2	1.05	1.05	90.35	0 / 0	517	516	0	High utilization
WS3	1.15	1.15	98.77	0 / 0	516	515	0	Near-full capacity
WS4	0.62	0.62	53.18	0 / 0	515	514	0	Underutilized
WS5	0.87	0.87	74.53	0 / 0	514	514	0	Moderate utilization
WS6	0.95	0.95	81.24	0 / 0	514	513	0	Stable throughput
WS7	0.8	0.8	68.31	0 / 0	513	512	0	Balanced processing
WS8	1.08	1.08	92.09	0 / 0	512	511	0	High utilization
WS9	0.63	0.63	53.66	0 / 0	511	511	0	Underutilized
WS10	0.8	0.8	68.02	0 / 0	511	510	0	Moderate activity
WS11	1.2	11.02	98.48	8.36 / 17	510	492	4,831	Secondary bottleneck
WS12	1.14	1.14	93.37	0 / 0	492	491	0	High utilization
WS13	1	1	81.76	0 / 0	491	490	0	Efficient processing
WS14	1.05	1.05	85.7	0 / 0	490	489	0	High utilization
WS15	0.8	0.8	65.2	0 / 0	489	489	0	Moderate utilization
WS16	0.8	0.8	65.08	0 / 0	489	488	0	Balanced throughput

The comparative summary of findings from the simulation are shown in Table 7.

Table 7. Comparative summary of workstation allocations

Metric	**RPW and GA**	**DE**	**Assessment**
Total Workstations	18	16	Fewer WS in DE, more compact
Main Bottleneck	WS2 – 99.9% utilization, Avg queue = 39.03, wait = 20,319 min	WS1 – 100% utilization, Avg queue = 41.38, wait = 21,341 min	Earlier in DE, bottleneck moved with similar intensity.
Secondary Bottleneck	None explicitly indicated	WS11 – 98.48% utilization, Avg queue = 8.36, wait = 9.82 min	Present in DE
High Utilization Stations	WS2, WS4, WS7, WS10, WS13, WS14, WS15	WS1, WS2, WS3, WS8, WS11, WS12, WS14	More evenly distributed in DE
Underutilized Stations	WS3, WS6, WS8, WS17	WS4, WS9	Fewer underutilized units in DE
Smooth Flow Stations	WS7, WS13, WS16, WS18	WS7, WS13, WS15, WS16	Similar performance
Total Output	509	488	Higher output in RPW and GA
Total m/c	23	23	Same Quantity
Efficiency	71%	77%	Higher efficiency in DE

Metric	RPW and GA	DE	Assessment
Predecessor constraint	No violation	Eight precedence violations were identified: 1.07–1.06, 3.01–6.03, 7.01–9.01, 7.01–9.02, 8.01–7.02, 6.01–9.03, 6.02–6.01, and 5.01–8.03.	DE violates predecessor constraint

6. Conclusion

The purpose of this study was to develop an automated method for garment layout optimization and Workstation allocation using Ranked Positional Weight (RPW) and Basic Pitch Time (BPT), and employ the Genetic Algorithm (GA) and Differential Evolution (DE) to perform assembly line balancing. The findings emphasize a balance between performance optimization and controlling operational constraints in line balancing. The application of DE led to a more compact layout with only 16 Workstations, compared to 18 in the Genetic Algorithm configuration, improving line efficiency by 5%. This result supports the usage of evolutionary algorithms to achieve better SMV variance and faster convergence, showcasing DE's ability to reach near-optimal load distributions more rapidly than GA.

Lower Workstations in the DE layout further ensure lower space and equipment costs, though it may also lead to more intense workloads at individual stations. While fewer Workstations typically indicate a more compact arrangement, each station may handle a greater portion of the workload. This tradeoff was also observed by Karatepe Mumcu (2024), who applied multiple heuristic methods to balance lighting assembly lines. Moreover, GA has multiple underutilized stations (e.g., WS3, WS5, WS8, WS14), indicating potential inefficiencies. DE, on the other hand, reports fewer underutilized stations. Furthermore, GA achieves 71%-line efficiency, while DE reaches 76%, reflecting a 5% relative improvement for DE. This higher efficiency is linked to better workload assignment and reduced station idling. Similar conclusions were drawn by Bappy et al. (2019), who used work-sharing line balancing in the apparel industry and reported improved labor efficiency by minimizing idle time. Moreover, DE converges in about 45 iterations in our model compared to 60 iterations in the GA layout, making it particularly well-suited to scenarios requiring swift re-optimization (e.g., rapid style changes or sudden demand shifts). GA can be advantageous in highly constrained or complex environments, but it demands a bit more computational time. Both GA and DE reduce bottlenecks compared to an unoptimized baseline. DE shifts the main bottleneck to an earlier station (WS1). However, the simulation results show that DE has more bottleneck workstation than GA, indicating that DE does not reduce bottleneck severity despite achieving fewer workstations and a more compact layout. DE's ability to reduce stations without drastically compromising flow can be especially beneficial when manufacturing space is limited or costly. GA's thorough search might be beneficial when predecessor constraints are tight. This comparative analysis between DE and GA is consistent with other studies (Tran et al., 2022; Tušar & Filipič, 2007). However, mild violations in predecessor constraints suggest that tighter constraint handling or additional refinement might be necessary before industrial implementation. Pearce et al. (2019) mentioned strict adherence to sequencing constraints in generalized ALB formulations. While previous studies achieved efficiency improvements using heuristic methods alone (Huskhazrin Kharuddin et al., 2020; Karatepe Mumcu, 2024; Kayar & Akyalçin, 2014), our integration of evolutionary algorithms provides a new and unique way by combining RPW-based heuristics with advanced optimization.

This hybrid approach of the study allows for more concrete search within the solution space and more aggressive consolidation of Workstations. Based on the findings, we conclude that DE provides a more efficient and compact solution for line balancing but may require refinement to fully satisfy the constraints. Whereas, GA maintains all the constraints, but at the cost of slower convergence and underutilized workstations.

References:

Achmadi, F., Harsanto, B. and Yunani, A., "Improvement of assembly manufacturing process through value stream mapping and ranked positional weight: An empirical evidence from the defense industry," *Processes*, vol. 11, no. 5, Art. no. 1334, 2023, doi: 10.3390/pr11051334.

Ahamed, Md. S., Farin, S., Suriya, K. S., Tabassum, T. and Choudhury, N. A., "Optimizing supplier order sizes to mitigate the bullwhip effect in a food supply chain: A case study based approach," *Proc. Int. Conf. Ind. Eng. Oper. Manag. (IEOM)*, 7th Bangladesh Int. Conf., Bangladesh, 2024, doi: 10.46254/BA07.20240182.

Ahmed, T., Sakib, N., Hridoy, R. M. and Shams, A. T., "Application of line balancing heuristics for achieving an effective layout: A case study," *Int. J. Res. Ind. Eng.*, vol. 9, no. 2, 2020, doi: 10.22105/riej.2020.234612.1134.

Akhter, S., Rutherford, S. and Chu, C., “Exploring the system capacity to meet occupational health and safety needs: The case of the ready-made garment industry in Bangladesh,” *BMC Health Serv. Res.*, vol. 19, Art. no. 435, 2019, doi: 10.1186/s12913-019-4291-y.

Arefin, Md. S., Roy, I., Chowdhury, S. and Alam, Md. S., “Employer safety obligations, safety climate, and safety behaviors in the ready-made garment context in Bangladesh,” *J. Safety Res.*, vol. 83, pp. 238–247, 2022, doi: 10.1016/j.jsr.2022.08.020.

Bappy, M. M., Musa, A. and Hossain, F., “Productivity improvement through line balancing: A case study in an apparel industry,” *Global Sci. J.*, vol. 7, no. 2, 2019.

Battaïa, O. and Dolgui, A., “A taxonomy of line balancing problems and their solution approaches,” *Int. J. Prod. Econ.*, vol. 142, no. 2, pp. 259–277, 2013, doi: 10.1016/j.ijpe.2012.10.020.

Bongomin, O., Mwasiagi, J. I., Nganyi, E. O. and Nibikora, I., “Improvement of garment assembly line efficiency using line balancing technique,” *Eng. Rep.*, vol. 2, no. 4, Art. no. e12157, 2020, doi: 10.1002/eng2.12157.

Boysen, N., Schulze, P. and Scholl, A., “Assembly line balancing: What happened in the last fifteen years?” *Eur. J. Oper. Res.*, vol. 301, no. 3, pp. 797–814, 2022, doi: 10.1016/j.ejor.2021.11.043.

Chen, J. C., Chen, C.-C., Su, L.-H., Wu, H.-B. and Sun, C.-J., “Assembly line balancing in garment industry,” *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10073–10081, 2012, doi: 10.1016/j.eswa.2012.02.055.

Chowdhury, M., Ahmed, R. and Yasmin, M., “Prospects and problems of RMG industry: A study on Bangladesh,” *Res. J. Finance Account.*, vol. 5, no. 7, pp. 103–118, 2014.

Dolgui, A. and Proth, J.-M., “Assembly line balancing: Conventional methods and extensions,” *IFAC Proc. Vol.*, vol. 46, no. 9, pp. 43–48, 2013, doi: 10.3182/20130619-3-RU-3018.00644.

Ghosh, S. and Gagnon, R. J., “A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems,” *Int. J. Prod. Res.*, vol. 27, no. 4, pp. 637–670, 1989, doi: 10.1080/00207548908942574.

Hazır, Ö. and Dolgui, A., “A review on robust assembly line balancing approaches,” *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 987–991, 2019, doi: 10.1016/j.ifacol.2019.11.323.

Hosen, M. D., “Reduction of garments bottleneck processing time on the sewing line of the garments industries,” *J. Textile Sci. Fashion Technol.*, vol. 5, no. 3, 2020, doi: 10.33552/JTSFT.2020.05.000611.

Huskhazrin Kharuddin, M., Fadzli Ramli, M. and Hafiz Masran, M., “Line balancing using heuristic procedure and simulation of assembly line,” *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 774–782, 2020, doi: 10.11591/ijeecs.v17.i2.pp774-782.

Islam, S. M., Das, S., Ghosh, S., Roy, S. and Suganthan, P. N., “An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 482–500, 2012, doi: 10.1109/TSMCB.2011.2167966.

Islam, T. and Halim, Md. A., “Impact of ready-made garments (RMG) industries and sustainability: Perspective of the pandemic period in developing country,” *Cleaner Eng. Technol.*, vol. 11, Art. no. 100567, 2022, doi: 10.1016/j.clet.2022.100567.

Karatepe Mumcu, Y., “Solution approach using heuristic and artificial neural networks methods in assembly line balancing problems: A case study in the lighting industry,” *Heliyon*, vol. 10, no. 5, Art. no. e26950, 2024, doi: 10.1016/j.heliyon.2024.e26950.

Katiraee, N., Calzavara, M., Finco, S., Battaïa, O. and Battini, D., “Assembly line balancing and worker assignment considering workers’ expertise and perceived physical effort,” *Int. J. Prod. Res.*, vol. 61, no. 20, pp. 6939–6959, 2023, doi: 10.1080/00207543.2022.2140219.

Katoch, S., Chauhan, S. S. and Kumar, V., “A review on genetic algorithm: Past, present, and future,” *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.

Kayar, M. and Akyalçin, Ö. C., “Applying different heuristic assembly line balancing methods in the apparel industry and their comparison,” *Fibres Textiles East. Eur.*, vol. 22, pp. 8–19, 2014.

Micieta, B. and Stollmann, V., “Assembly line balancing,” in *DAAAM Int. Scientific Book 2011*, Vienna, Austria: DAAAM Int., 2011, doi: 10.2507/daaam.scibook.2011.21.

Noor, A., Saeed, M. A., Ullah, T., Uddin, Z. and Ullah Khan, R. M. W., “A review of artificial intelligence applications in apparel industry,” *J. Textile Inst.*, vol. 113, no. 3, pp. 505–514, 2022, doi: 10.1080/00405000.2021.1880088.

Pearce, B. W., Antani, K., Mears, L., Funk, K., Mayorga, M. E. and Kurz, M. E., “An effective integer program for a general assembly line balancing problem with parallel workers and additional assignment restrictions,” *J. Manuf. Syst.*, vol. 50, pp. 180–192, 2019, doi: 10.1016/j.jmsy.2018.12.011.

Ponnambalam, S. G., Aravindan, P. and Naidu, G. M., “A comparative evaluation of assembly line balancing heuristics,” *Int. J. Adv. Manuf. Technol.*, vol. 15, no. 8, pp. 577–586, 1999, doi: 10.1007/s001700050105.

Rahman, A., Acharjee, P., Sharif, T., Ishraq-Ul-Hoque, M. and Choudhury, N. A., “A comparative evaluation of line balancing methods to enhance sewing line performance,” *Proc. Int. Conf. Ind. Eng. Oper. Manag. (IEOM)*, 7th Bangladesh Int. Conf., Bangladesh, Dec. 2024, doi: 10.46254/BA07.20240062.

Rahnamayan, S., Tizhoosh, H. R. and Salama, M. M. A., “Opposition-based differential evolution,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, 2008, doi: 10.1109/TEVC.2007.894200.

Schoenfelder, J., Heins, J. and Brunner, J. O., “Task assignments with rotations and flexible shift starts to improve demand coverage and staff satisfaction in healthcare,” *J. Sched.*, vol. 28, no. 3, pp. 329–353, 2025, doi: 10.1007/s10951-025-00838-z.

Tareque, M. A., Islam, N. and Roy, S., “Increasing efficiency: Case study of ready-made garments in Bangladesh,” *Int. J. Eng. Comput. Sci.*, vol. 9, no. 6, pp. 25085–25101, 2020, doi: 10.18535/ijecs/v9i06.4503.

Tran, H.-N., Phan, G. T. T., Do, Q. B. and Tran, V.-P., “Comparative evaluation of the performance of improved genetic algorithms and differential evolution for in-core fuel management of a research reactor,” *Nucl. Eng. Des.*, vol. 398, Art. no. 111953, 2022, doi: 10.1016/j.nucengdes.2022.111953.

Tušar, T. and Filipič, B., “Differential evolution versus genetic algorithms in multiobjective optimization,” in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, Berlin, Heidelberg: Springer, pp. 257–271, 2007, doi: 10.1007/978-3-540-70928-2_22.

Biographies

Kazi Fahim is a Production Engineer at Snowtex Group and a graduate of Industrial and Production Engineering from the National Institute of Textile Engineering and Research (NITER). With experience in enhancing manufacturing efficiency and process performance, his research interests include industrial system optimization, production and operations management, lean manufacturing, industrial automation and the development of hybrid, adaptive heuristic and metaheuristic algorithms for solving complex industrial problems.

Sharafat Hossain is a Production Engineer at Snowtex Group, where he works on sustainable manufacturing and process optimization projects. He earned his B.Sc. in Industrial and Production Engineering from Ahsanullah University of Science and Technology (AUST), Dhaka. His research interests include workstation design optimization, lean and sustainable manufacturing, industrial automation, and data-driven production improvement.

Joyanta Debnath is a Production Engineer at Snowtex Group with a B.Sc. in Industrial and Production Engineering from Ahsanullah University of Science and Technology (AUST). His professional focus centers on enhancing manufacturing performance through industrial system optimization and data-driven decision-making. His research interests include advancements in production planning, resource allocation, and continuous process improvement.

Nurul Ahad Choudhury is a PhD student and Graduate Research Assistant in the Department of Industrial Engineering and Management at Oklahoma State University. His research interests are cognitive engineering and decision making, human factors engineering, and renewable energy. Previously, he served as a Lecturer at the National Institute of Textile Engineering and Research. Nurul’s current research focuses on improving cognitive performance among professionals working in crisis situations.