Parallel Assembly Line Balancing With Human Robot Collaboration

TaranpreetSingh H Rababi, Shreya Hiremath, Jayalaxmi G N, Sachin Karadgi
KLE Technological University
Hubballi 580031, Karnataka, India
Email: {01fe23mcs009, 01fe23mcs014, jaya_gn, sachin.karadgi}@kletech.ac.in

***Abstract*—Human-robot collaboration (HRC) will indeed transform production paradigms in intelligent manufacturing. The aim of this research is to investigate the application and merits of HRC in parallel conveyors and to identify how it may contribute to enhanced productivity, resource use, and operational flexibility. The main issue being addressed is incorporating human-robot collaboration features into the Parallel Assembly Line Balancing Problem (PALBP), which is now referred to as PALBP-HRC. These collaboration modes allow for both parallel and cooperative human-unattended robot operation while enabling the optimization of tasks allocation.**

For some problems of that hybrid nature associated with PALBP-HRC, a genetic algorithm is proposed to minimize cycle time along with optimal task assignment, keeping in mind the different complex restrictions such as precedence relations, zoning restrictions, and human-robot compatibility for tasks. GA also seeks to maximize benefits deriving from the distinct strengths of humans and robots by using a proper form of task allocation.

The experiments show how the proposed GA has demonstrated its capability in solving PALBP-HRC. The cycle time has been significantly reduced with most improved task assignment and flexibility in operations. Thus, the results indicate that human-robot collaboration has been seamlessly included in a parallel assembly line to emphasize not only productivity enhancement but also adaptability in dynamic production environments.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A. Assembly Line Balancing Problem (ALBP)

The assembly Line Balancing problem (ALBP) is a crucial optimization venture in manufacturing and production structures. It entails distributing obligations amongst workstations along an meeting line to obtain specific objectives, which includes minimizing cycle time, reducing the range of workstations, or maximizing performance, whilst adhering to various constraints like undertaking priority, zoning, and workload stability. green meeting line balancing is crucial for enhancing productiveness, lowering expenses, and making sure smooth operations in industries starting from automotive to electronics. The trouble is inherently combinatorial and regularly solved the usage of optimization strategies, including heuristic, meta-heuristic, or exact techniques, to address its complexity and ranging industrial necessities. meeting line balancing allows in optimizing labor, assets, and device usage by way of cautiously distributing duties based totally on time requirements and priority. Balancing properly can cause fee financial savings, quicker manufacturing, and higher throughput. however, choppy project intervals, aid obstacles, and variability in performance complicate the balancing process.

Fig. 1. TV Manufacturing Assembly line [1]

Fig. 2. Car Manufacturing Assembly line [2]

Fig 1 and Fig 2 are some of the visually examples to have an overview of how an assembly-line looks like and work both humnas and robots.

B. Parallel Assembly Line Balancing Problem (PALBP)

Human-robot collaboration (HRC), as an rising manufacturing mode, has garnered significant interest inside the context of the growing clever manufacturing. Parallel meeting strains, they perform with numerous traces jogging concurrently. (for instance in car manufacturing wherein a car may have frame assembly, painting and engine installation simultaneously.)

those parallel lines can be balanced the use of multi-line stations in which obligations for adjacent traces are completed together. This situation is known as the parallel assembly line balancing problem (PALBP). the combination of collaborative robots into parallel assembly traces can in addition increase the performance of the assembly device called PALBP- HRC. Parallel line balancing challenges contain allocating obligations correctly throughout parallel lines in such a manner that no line ends up turning into a bottleneck and every stays in stability regarding paintings distribution and cycle time. This entails handling the priority constraints of the mission and balancing the workload such that no unmarried line acts as a bottleneck, at the same time as all strains run at top performance. This process seeks to ensure a minimized downtime, decreased put off and a smooth glide of productions in all traces. Parallel line balancing may be very sizeable in which there is a huge demand for production and only a single line can't make to the goal manufacturing hence necessitates multiple strains to cater to the demand. This balances the useful resource usage by using providing higher throughput and lets in most flexibility in manufacturing.

Fig. 3. Human robot collaboration Assembly line [3]

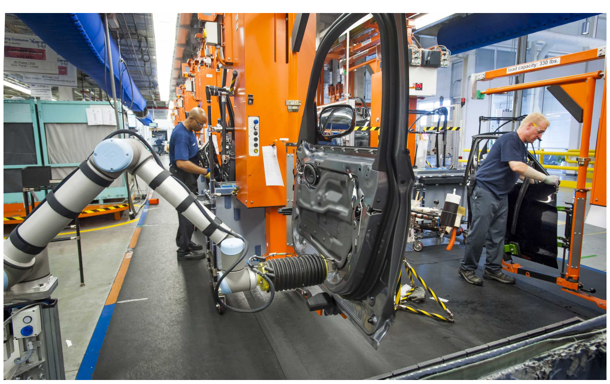

Fig. 4. Human robot collaboration Assembly line [4]

Figs. 3 and 4 give the visual real world uses of human and robot collaboration in the industry to manufacture products.

II. Literature Survey

The hassle of assembly line balancing has been drastically studied with several optimization strategies proposed to cope with the complex troubles in this vicinity. those variety from traditional strategies through blended-integer programming and heuristic algorithms for arriving at an most suitable or near-highest quality solutions challenge to predefined constraints. This, these days, meta-heuristic techniques like genetic algorithms, simulated annealing, and particle swarm optimization have come within the highlight due to the ability of solving multi-objective issues. on this segment, an in depth evaluation of to be had works in assembly line balancing is supplied with emphasis on their methodologies, obstacles, and applicability to parallel and bendy manufacturing structures

A new paralleled assembly lines (two to three) run side by side and structured into human-robot collaboration balancing, for example, with case study of HRC, PALBP-HRC-II. Nourmohammadi et al. (2023) [5] present assembly lines endowed with sensing and interaction technologies for optimizing three maximization and minimization objectives for workstation (NS) minimization concerning cycle time (CT), CT minimization for a given NS, and overall cost minimization regarding stations, operators, and robot power consumption. Fathi et al. (2024) [6] present a number of mathematical models for problems of standard and collaborative robot balancing, with six models: SALBP-I, SALBP-II, RALBP-I, RALBP-II, ALBP-HRC-I, and ALBP-HRC-II, and a comparison of major parameters, decision variables, and constraints. Weckenborg et al. (2019) [7] propose a model where a number of co-robots assign themselves to stations and distribute workloads between human workers and robots for minimum cycle time purposes. Mao et al. (2023) [8] define and develop an assembly line type as U, for these cases of human-robot collaboration (UALBP-HRC) and provide a mixed-integer programming model aimed at minimizing cycle time and upgrading techniques such as tight bounds and initial solutions. These studies submitted over human-robot collaborative assembly line balancing are work-optimizedness, minimized cycle time, and cost-effectiveness.

III. Methodology

The project used a genetic algorithm to maximize scheduling in a lineless mobile assembly system. Modeling the problem was part of the methodology. To account for uncertainties, design an adaptive genetic algorithm for scheduling and simulation. It is possible to address dynamic job arrivals. The approach is focused on flexibility, less delay, and all-round efficiency.

A. General Flow Diagram

A genetic set of rules is genuinely a search and optimization approach, stimulated through the standards of natural choice and evolutionary biology. In different phrases, it mimics the technique of evolution, evolving solutions to complicated optimization problems over successive generations through mechanisms like selection, crossover, and mutation.

Fig 5 shows the flow chart that describes the general architecture of a genetic algorithm (GA) and the algorithm ALG 1 shows the detailed algorithm of GA, that typically involves the following steps:

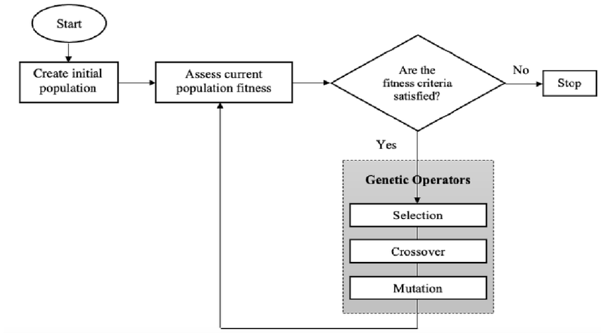

Fig. 5. General flow Diagram [9]

Input: Initial population P, Fitness function f, Termination condition T, Selection operator S, Crossover operator C, Mutation operator M, Population size N
Output: Best solution x^*
Initialize population P randomly with size N;
while *termination condition* T *not satisfied* **do**
    foreach *individual* x *from* P **do**
        Evaluate $f(x)$;
    end
    Selection: Select individuals from P for the mating pool according to their fitness using operator S (such as roulette wheel, tournament selection);
    Crossover: Apply the crossover operator C over pairs in the mating pool, with crossover probability p_c, to generate offspring;
    Mutation: Introduce diversity to the generated offspring by applying mutation operator M with a mutation probability of p_m;
    Replacement: Create the new population combining offspring with the fittest individuals from the current population (elitism, for example);
    Update: P is set to the new population;
end
Find the best solution x^* among those in P by evaluating it using $f(x)$.
return x^*

Algorithm 1: Genetic Algorithm (GA)

B. High Level Design

Fig 6 shows the high flow of a meta-heuristic algorithm for the solution of optimization problems.

1) Input:
   - At this step, problem data is given to the meta-heuristics algorithm.
   - It usually contains:
     - Define the problem (e.g., a scheduling task, a traveling salesman problem, or a resource allocation problem).
     - Initial solution(s) or search space parameters.

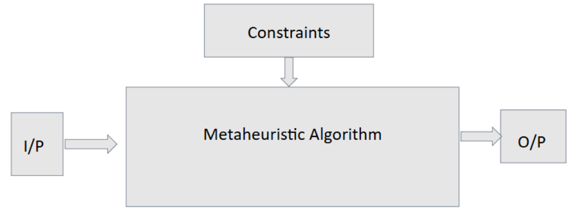

Fig. 6. High-Level Design

     - Algorithm-specific parameters such as population size (if population-based), iteration limits, or step size.

2) Constraints:
   - Constraints are the rules or conditions that are satisfied for a solution to hold a validity.
   - they are defining the feasibility of solutions in the search space. Constraints, for instance, inside the case of scheduling
     - problems: ”No two events can overlap”.
     - In route optimization: ”A vehicle cannot exceed its maximum capacity”.
     - Task assignments: ”A task must respect precedence relationships”.
   - These constraints guide the meta-heuristic algorithm in avoiding invalid solutions while looking for an optimal one.

3) Meta-Heuristic Algorithm:
   - A core portion in this diagram is the optimization which in fact derives from the meta-heuristic algorithm.
   - Meta-heuristic deals with high-level strategies that enable solving highly complicated optimization problems by intelligently exploring the solution space.
   - It balances two critical sides:
     - Exploring: Searching diverse regions of the solution space.
     - Exploiting: Refinement of the current best solutions.
   - Examples of Meta-heuristic Algorithms:
     - Genetic Algorithm (GA): which derives its line from the natural selection and evolution.
     - Simulated Annealing (SA): which is similar to the cooling process of metals.
     - Particle Swarm Optimization (PSO): which derives its line from the social behavior of birds or fish.
     - Ant-Colony-Optimizations (ACO): which is an imitation of how ants find the shortest paths.
     - Tabu Search: uses memory to avoid revisiting previous solutions.

4) Output:
   - This is the final output as produced by the optimization process.
   - The outputs are between:
     - The best solution found using the meta-heuristic algorithm within the confines of constraints.
     - Some key performance indicators such as minimized cost, maximized profit, optimized schedule.

For the proposed GA the main techniques used are given as follows,

1) **Initial Population:** Generation of initial individuals (solutions) related to task sequence respect precedence constraints. Then shuffle task assignments separately for Line 1 and Line 2. Checks whether the shuffled assignment for an individual satisfies the precedence constraints. Only "valid" individuals are accepted into the population.
2) **Tournament Selection:** Individuals of the population are chosen using tournament selection. Randomly choose a number of individuals (tournament_size) from the population. The one with the best (lowest) fitness is selected. This is repeated until a new selected population is formed.
3) **Single Point Crossover:** Carries out a single-point crossover but with offspring that are maintaining the precedence constraints. Cuts parents at a randomly defined crossover point and interchanges parts to create offspring. Both offsprings are checked against the precedence. If offspring is invalid then goes back to the original parent.
4) **Swap Mutation:** It exercises swap mutation (swapping two tasks) with respect to most precedence constraints. Tends to try up to 10 random swaps. Returns the first mutated form that is valid. In case no valid mutation is obtained, it returns the original.
5) **Elitism:** Elite individuals with fitness values selected will be carried over unchanged to the next generation. Sorts individuals by the their fitness scores. Returning the top-performing individual.

With addition to the GA, the follwoing constraints are applied to make sure that the algorithm behaves as intended and gives the optimal solution.

1) **Assignment of Task to Robot, Human or Human-Robot:** Assuming that the collaboration would speed up the processing, one would expect that the load would be reduced to 70% in a station with a robot. And the bigger task are assigned to robots whereas the small tasks to humans.
2) **Zoning Constraint:** Zoning refers to the tasks that can be accomplished together. Applies zoning to optimize positive/negative zoning compatibility between tasks. If tasks are precedence constrained, they are positively zoned (should be together). If not, they are negatively zoned (should never be together). Any zoning violations result in tasks being swapped.

IV. Results and Discussions

The consequences of the proposed genetic algorithm are analyzed to check effectiveness in optimizing parallel meeting line balancing. The overall performance of all of the check instances is in comparison concerning cycle time, undertaking allocation efficiency, and constraint pleasure. This evaluation is presented through graph, chart, and Gantt diagrams for illustrating mission assignments and overall gadget performance. This phase also gives a essential dialogue at the found results, with a focus on strengths and limitations of the algorithm.

TABLE I
Cycle Times and Zoning percentage

Dataset Label	Cycle Time(sec)	Positive Zoning%	Negative Zoning%
JACKSON	23.1	25.714286	74.285714
jaeschke	18.2	22.058824	77.941176
MANSOOR	88.2	0.000000	100.000000
mertens	15.0	30.769231	69.230769
MITCHELL	53.0	27.317073	72.682927
BOWMAN	38.5	22.857143	77.142857

The Table I contains information on Cycle Time and Zoning Constraint Satisfaction within different datasets. Cycle Time shows a broad spread with Mansoor at the peak (88.2) and Mertens at the bottom (15.0). High cycle times usually indicate inefficiencies, while a low cycle time suggests a better time-distribution of tasks.

In Zoning Compliance, Mansoor does not get a positive 0% zoning and a 100% violation, while Mertens has the best percentage of positive zoning so far, which is at 30.77%. Most other datasets lie at zoning compliance of about 22-27%, which indicates potential improvement. Lower cycle time does not lower the extent of zoning compliance: saw in the example of Mitchell, which has cycle time (53.0) and zoning compliance (27.3%).

To optimize performance, it would be important to reduce the cycle time and improve zoning constraints.

A. Cycle Times

Such a thing is often witnessed in graphs. Different bars describe different cycles for different folders, where each bar gives the time taken by the task processed through that folder. The x-axis refers to folder names, and the y-axis describes the respective cycle times in numerical units. Fig 7 and Fig 8 shows the cycle times of all dataset.

Observation in Detail:

- **Longest Cycle Time:** This folder, MANSOOR, contains the longest vertical bar in the graph, it reflects the highest cycle time, which is over 90 units, indicating that the amount of time taken to work on tasks or operations related to this folder is substantially longer than that of other folders.

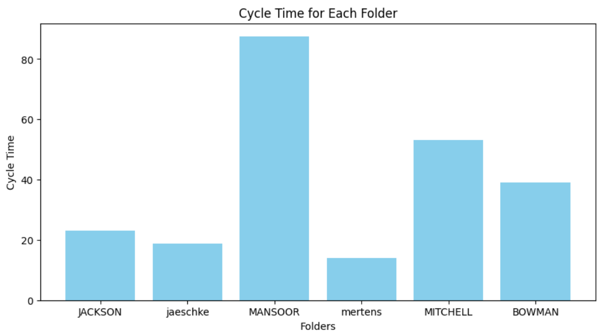

Fig. 7. Cycle Time Bar Graph

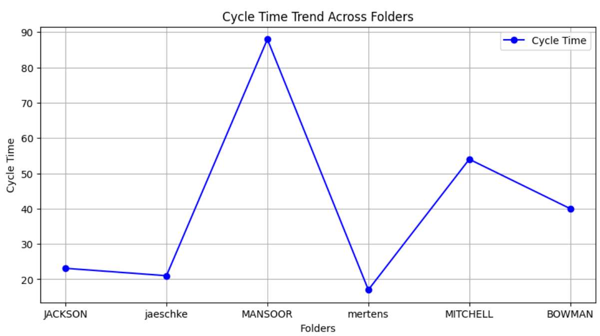

Fig. 8. Cycle Time Line Graph

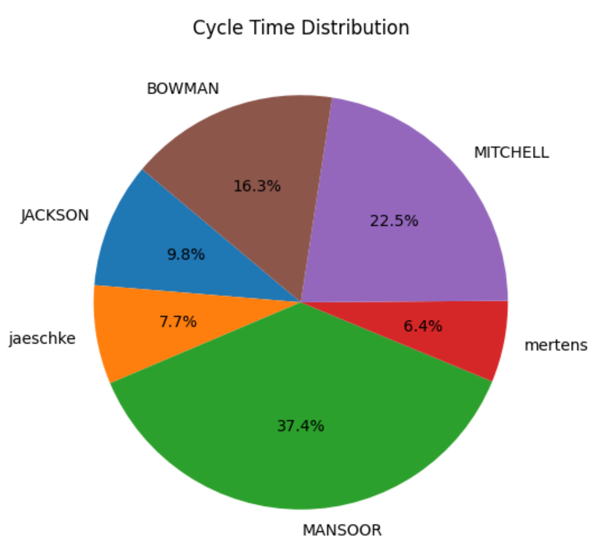

Fig. 9. Cycle Time Pie Graph

- **Shortest Cycle Time:** This is the folder of mertens, with the shortest height of the bars by which it is indicated as this is the minimum cycle time of all, approximately 10 units, indicating that the tasks associated with this folder can be done much faster than any other folder's tasks.
- **Intermediate Cycle Times:**
  - The average points for MITCHELL and BOWMAN are in between, falling somewhere between 40 and 60 units. These two folders seem to be balanced, neither being the fastest at task completion nor being the slowest.
  - JACKSON and jaeschke have pretty much the same cycle times, being in the range of 20 to 30 units, which shows that the processing duration is between low and average.
- **Variation Across Folders:** The graph here shows variation in cycle times considerably from folder to folder, meaning they have a different demand in processing or complexity in the type of tasks tackled. The difference between the maximum and minimum cycle time is huge, with MANSOOR being more than 9 times slower than mertens.
- **Distribution Pattern:** Most folders (4 out of 6) will have cycle times less than 50 units, except for MITCHELL and BOWMAN who approach or go above that threshold. Hence, most tasks or data sets are quick to process, while only a few names take much longer.

B. *Zoning Constraint*

The zoning percentages amounts per folder have been considered in this graph. The X-axis refers to the name of the folder, while the Y-axis shows the zoning constraint percentages of 0% - 100%. Each folder has its bar presented in two- the bar showing a positive zoning percentage sharing green color and another bar for negative zoning percentage in red color. Fig 10 and Fig 11 shows the zoning constraint percentages.

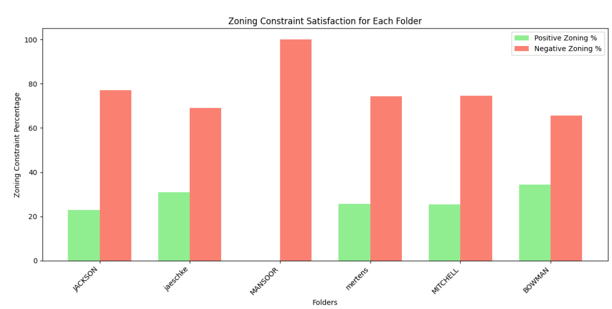

Fig. 10. Zoning Constraint Percentage Bar Graph

In Detailed Observations:

- **Positive Zoning (Green Bars):**
  - Positive zoning percentages have been low throughout the folders, being below 30% for most of the criteria.
  - Folders jaeschke, mertens, and BOWMAN are slightly ahead in zoning positive percentages with respect to JACKSON, MITCHELL, and MANSOOR. However even these are too deficient to show much inclination towards positive zoning.
- **Negative Zoning (Red Bars):**

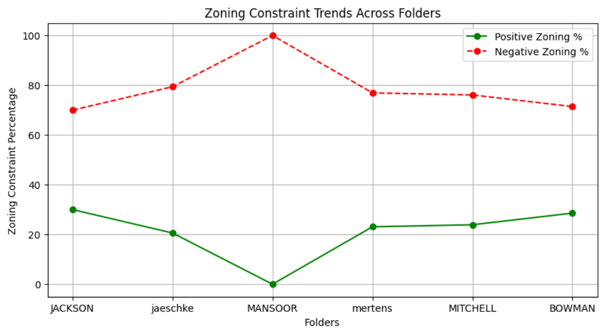

Fig. 11. Zoning Constraint Percentage Line Graph

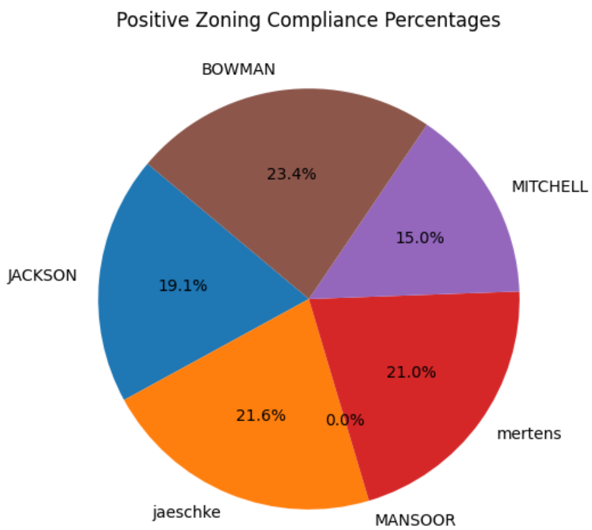

Fig. 12. Positive Zoning Constraint Percentage

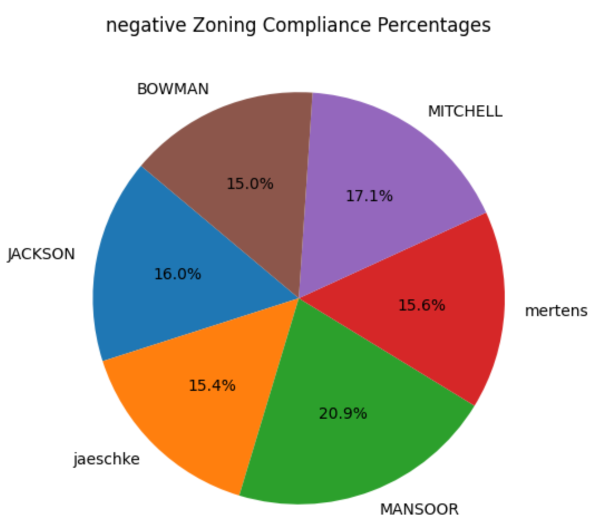

Fig. 13. Negative Zoning Constraint Percentage

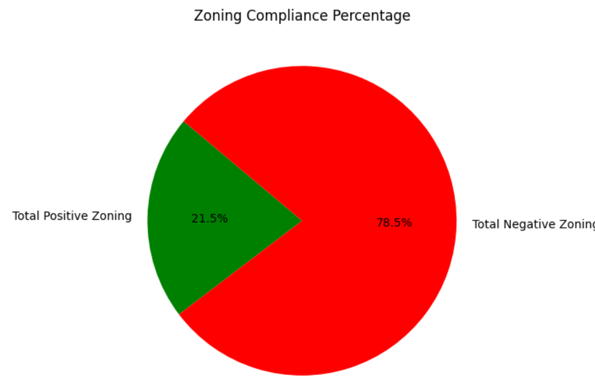

Fig. 14. Zoning Constraint Percentage Pie Graph

- Negative zoning tops all folders with values greater than 60% and at some point may even reach 100% (like in the case of MANSOOR).
- The place that holds the highest at negative-zoning is MANSOOR, who has the 100% at representing all the negative-zoning-qualified violations in this folder.
- Then there is JACKSON ranking high in negative zoning with their values ranging approximately between 70% and 90%.

- **Comparative Results of the Above:**
  - A clear distinction is visible between green and red bars of all folders, which shows imbalance in it. All positive percentages seem to pale against negative percentages.
  - Thus, it indicates that negative zoning infringements have been general to all folders, whereas positive zoning constraints with very little compliance are met.
- **Folder-Specific Analysis:**
  - It cannot be considered that this folder has some positive zoning since this folder has an entirely negative zone; that is 0% positive with a red bar taking the complete height at 100%.
  - The rest, jaeschke and BOWMAN, have around 20-30% higher positive zoning compared to the other folders.

C. Assignment of Tasks to Stations

The created Gantt charts for each folder as a part of the assignment, which would give the visualization of the distribution of tasks at various stations. The Gantt charts would show how jobs are being assigned to different stations, with initiation timings and duration. These helped in judging the efficiency and timeline possibility of the assembly line operation. A clear and visual representation of the task assignment process is an important tool to measure the feasibility and efficiency in the direction and allocation of tasks throughout the assembly line. Fig 15, Fig 16, Fig 17, Fig 18, Fig 19, Fig 20 shows the overall assignments of tasks

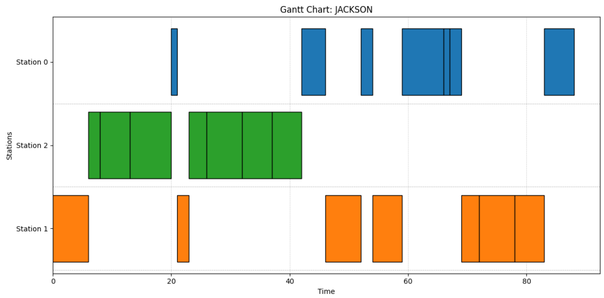

Fig. 15. Task Assignment of Jackson Data

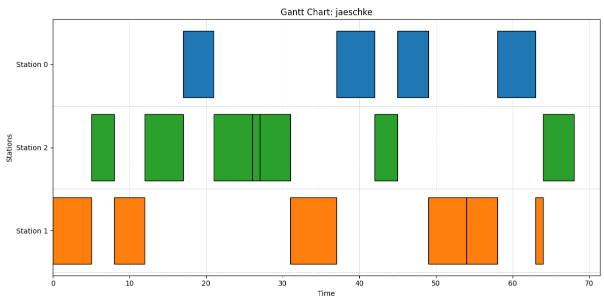

Fig. 16. Task Assignment of jaeschke Data

to stations respectively.

Important observations:

- **X-axis (Time):** That is the time axis, during which the tasks will be performed.
- **Y-axis (Stations):** The stations are listed on the Y-axis, and each one corresponds to different task assignment within the folder.
- **Bars:** Each bar in the Gantt chart describes a task that is assigned to an individual station, where:
- The left facet of the bar corresponds to the time the project had began. The duration of the bar indicates the duration of the mission at that station.
- The Gantt charts are colour-coded by station, every with a one-of-a-kind colour. This permits easy identity of obligations assigned to the identical station. The duties are marked with their assignment number (as an example, "project 1," "task 2"), and the legend could distinguish the different duties.

Folder-wise Analysis and Insights:

- **Task Distribution Across Stations:** The gantt charts in the folders show how tasks are distributed across various stations. For example, Folder A might get an even share of task distribution whereas Folder B will have some of the stations overloaded at certain times.
- **Task Scheduling and Duration:** The start and duration times of each task under each folder give a feeling of the well the tasks could be executed. A task could be at a station with a long duration, which could mean a bottleneck or improvement area.
- **Task Conflicts and Overlaps:** Gantt charts would show potential overlaps in work at a station. For instance, if two or more tasks are to start together at the same station,

Fig. 17. Task Assignment of MANSOOR Data

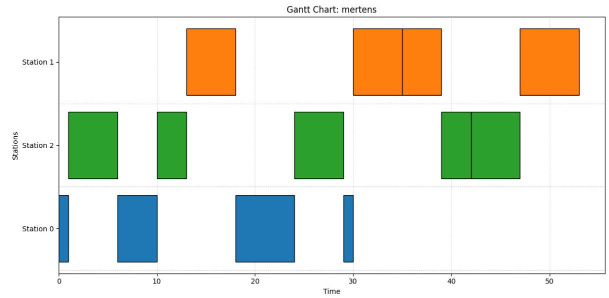

Fig. 18. Task Assignment of mertens Data

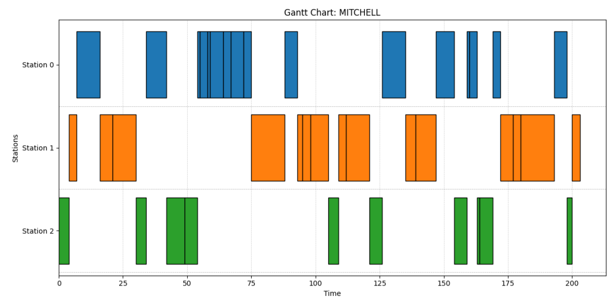

Fig. 19. Task Assignment of MITCHELL Data

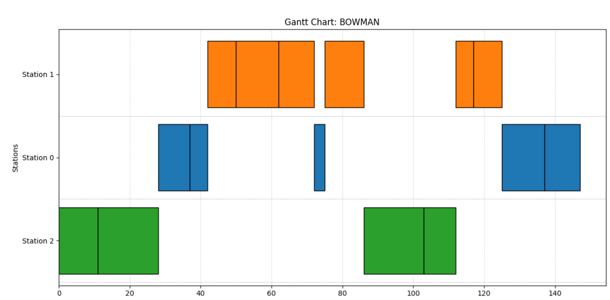

Fig. 20. Task Assignment of BOWMAN Data

then it indicates a possible conflict.

- **Optimization Opportunities:** Gantt charts can show where tasks can be rescheduled or where the work can be redistributed to improve the overall efficiencies of the assembly line. For instance, two tasks, which are so close to each other at one station, can be moved up or down from the scheduled time to avoid idle time or congestion.

V. CONCLUSION AND FUTURE WORK

The trouble lies within the allocation of responsibilities to stations, while the sub issues include the choice of processing alternatives for every of the responsibilities. A key factor is the zoning constraint, which lets in the coupling of two duties to be done on the identical station if they fall inside a pre-defined sector. this would allow the parallel execution of tasks such that it adds performance to the meeting line. Zoning constraints determine the challenge organization, i.e., it validates that a mission falls in a chosen zone so that the duties may be done concurrently, therefore nullifying useless time and increasing throughput. On a extraordinary notice, extending the idea of parallel lines with a cooperative format to distinct configurations, together with U-kind traces and -sided traces, will create a great opportunity for optimization. The mission was in finding out how exclusive layouts should have an effect on task assignment and whether or not or no longer line configurations could improve the overall line performance. extra static is the hassle concerning whether or not parallelism of the collaborative framework amongst exceptional varieties of lines can yield efficiency upgrades. this could include the interaction between multiple line types underneath one shared assignment allocation strategy. ultimately, as we strive to end up greater attentive to marketplace demands, the addition of combined fashions based totally on the above techniques remains a promising avenue. Merging mixed version production techniques with zoning constraints and cooperative line layouts would offer the agility now required to reply unexpectedly to changing marketplace conditions with out sacrificing performance across different project assignments.

REFERENCES

[1] "Tv manufacturing source." [Online]. Available: https://infinity-staffing.com/looking-for-work/

[2] "uto-firms-buoyed-by-festive-sales-unsure-of-sustained-demand-after-diwali." [Online]. Available: https://images.app.goo.gl/rWMBPusAUKS1j88L7

[3] "91862-human-robot-collaboration-comes-of-age." [Online]. Available: https://www.assemblymag.com/articles/91862-human-robot-collaboration-comes-of-age

[4] "innovative human robot cooperation in bmw group production." [Online]. Available: https://images.app.goo.gl/XuVzVu8CxnqGgfUp8

[5] A. Nourmohammadi, M. Fathi, and A. H. Ng, "Balancing and scheduling human-robot collaborated assembly lines with layout and objective consideration," *Elsevier*, 2023. [Online]. Available: https://doi.org/10.1016/j.cie.2023.109775

[6] M. Fathi, A. Sepehri, M. Ghobakhloo, M. Iranmanesh, and M.-L. Tseng, "Balancing assembly lines with industrial and collaborative robots: Current trends and future research directions," *Elsevier*, 2024. [Online]. Available: https://doi.org/10.1016/j.cie.2024.110254

[7] C. Weckenborg, K. Kieckhäfer, C. Müller, M. Grunewald, and T. S. Spengler, "Balancing of assembly lines with collaborative robots," *Springer*, 2019. [Online]. Available: https://doi.org/10.1007/s40685-019-0101-y

[8] Z. Mao, J. Zhang, K. Fang, D. Huang, and Y. Sun, "Balancing u-type assembly lines with human-robot collaboration," *Elsevier*, 2023. [Online]. Available: https://doi.org/10.1016/j.co

[9] F. Bagheri-Moghaddam, S. Banihashemi, R. Bakhshoodeh, J. Fort, and I. Navarro, "Biomimicry green façade: Integrating nature into building façades for enhanced building envelope efficiency," *SSRN*, 2023. [Online]. Available: https://www.researchgate.net/publication/372601856_Biomimicry_Green_Facade_Integrating_Nature_into_Building_Facades_for_Enhanced_Building_Envelope_Efficiency

[10] Z. Mao, J. Zhang, Y. Sun, K. Fang, and D. Huang, "Balancing parallel assembly lines with human-robot collaboration: problem definition, mathematical model and tabu search approach," *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 2024. [Online]. Available: https://doi.org/10.1080/00207543.2024.2356627

[11] H. Aguilar, A. García-Villoria, and R. Pastor, "A survey of the parallel assembly lines balancing problem," *Computers & Operations Research*, vol. 124, p. 105061, 2020. [Online]. Available: https://doi.org/10.1016/j.cor.2020.105061

[12] F. Araújo, F. B. Costa, M. Alysson, and C. Miralles, "Balancing parallel assembly lines with disabled workers," *European Journal of Industrial Engineering*, vol. 9, no. 3, pp. 344–365, 2015. [Online]. Available: https://doi.org/10.1504/EJIE.2015.069343

[13] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solution approaches," *International Journal of Production Economics*, vol. 142, no. 2, pp. 259–277, 2013. [Online]. Available: https://doi.org/10.1016/j.ijpe.2012.10.020

[14] N. Boysen, P. Schulze, and A. Scholl, "Assembly line balancing: What happened in the last fifteen years?" *European Journal of Operational Research*, vol. 301, no. 3, pp. 797–814, 2022. [Online]. Available: https://doi.org/10.1016/j.ejor.2021.11.043

[15] H. Çerçioğlu, U. Ozcan, H. Gökçen, and B. Toklu, "A simulated annealing approach for parallel assembly line balancing problem," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 24, no. 2, 2009.

[16] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1065–1075, 2013. [Online]. Available: https://doi.org/10.1109/TASE.2013.2274099

[17] P. Chutima, "Assembly line balancing with cobots: An extensive review and critiques," *International Journal of Industrial Engineering Computations*, vol. 14, no. 4, pp. 785–804, 2023. [Online]. Available: https://doi.org/10.5267/j.ijiec.2023.7.001

[18] P. Chutima and P. Jirachai, "Parallel u-shaped assembly line balancing with adaptive moea/d hybridized with bbo," *Journal of Industrial and Production Engineering*, vol. 37, no. 2-3, pp. 97–119, 2020. [Online]. Available: https://doi.org/10.1080/21681015.2020.1735544

[19] Z. A. cil, Z. Li, S. Mete, and E. Özceylan, "Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration," *Applied Soft Computing*, vol. 93, p. 106394, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2020.106394

[20] M. Dalle Mura and G. Dini, "Designing assembly lines with humans and collaborative robots: A genetic approach," *CIRP Annals*, vol. 68, no. 1, pp. 1–4, 2019. [Online]. Available: https://doi.org/10.1016/j.cirp.2019.04.006

[21] E. Dar-El, "Malb–a heuristic technique for balancing large single-model assembly lines," *AIIE Transactions*, vol. 5, no. 4, pp. 343–356, 1973. [Online]. Available: https://doi.org/10.1080/05695557308974922

[22] Y. Delice, E. K. Aydoğan, U. Özcan, and M. S. İlkay, "Balancing two-sided u-type assembly lines using modified particle swarm optimization algorithm," *4OR*, vol. 15, no. 1, pp. 37–66, 2017. [Online]. Available: https://doi.org/10.1007/s10288-016-0320-4

[23] G. R. Esmaeilian, S. Sulaiman, N. Ismail, M. Hamedi, and M. M. H. M. Ahmad, "A tabu search approach for mixed-model parallel assembly line balancing problem (type ii)," *International Journal of Industrial and Systems Engineering*, vol. 8, no. 4, pp. 407–431, 2011. [Online]. Available: https://doi.org/10.1504/IJISE.2011.041803

[24] F. Glover, "Tabu search–part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989. [Online]. Available: https://doi.org/10.1287/ijoc.1.3.190