Flexible Services and Manufacturing Journal
https://doi.org/10.1007/s10696-024-09533-1

Balancing and sequencing of mixed-model parallel robotic assembly lines considering energy consumption

Halenur Soysal-Kurt[1] · Selçuk Kürşat İşleyen[2] · Hadi Gökçen[2]

Accepted: 22 January 2024

Abstract
As technology advances, the integration of robots in the assembly line has become widespread. While robots offer numerous benefits, such as increased productivity and improved product quality, they also result in higher energy usage. Finding the optimal line balance while considering energy consumption is a challenging task in a robotic assembly line that produces multiple product models in a mixed sequence. This paper addresses the mixed-model parallel robotic assembly line balancing and model sequencing (MPRALB/S) problem. The objectives of this problem are to minimize cycle time and energy consumption. The authors have not found any existing research on this topic in the literature. To solve the MPRALB/S problem, a modified non-dominated sorting genetic algorithm II (MNSGA-II) is developed. Since there is no existing benchmark data for the MPRALB/S problem, new datasets are generated for this study. The MPRALB/S problem is illustrated through a numerical example. The performance of MNSGA-II is evaluated with non-dominated sorting genetic algorithm II (NSGA-II) and restarted simulated annealing through commonly used performance metrics including hypervolume ratio (HVR), ratio of non-dominated solutions (RP) and generational distance (GD). According to the results of the computational study, MNSGA-II outperforms NSGA-II in approximately 81% of the problem instances for HVR, 71% for RP, and 76% for GD. The results show that MNSGA-II is an effective approach for solving the MPRALB/S problem and achieves competing performance compared to other algorithms.

Keywords Mixed-model · Parallel · Robotic assembly line · Energy consumption · Balancing and sequencing problem · NSGA-II

✉ Halenur Soysal-Kurt
halenursoysal@osmaniye.edu.tr

[1] Department of Management Information Systems, Osmaniye Korkut Ata University, Osmaniye, Turkey

[2] Department of Industrial Engineering, Gazi University, Ankara, Turkey

1 Introduction

ALs are flow manufacturing systems that are composed of stations and a material handling system. Products move between consecutive stations to be assembled. Specific tasks are performed at each station within a certain cycle time. Assigning tasks to stations to achieve certain objectives within a set of constraints is an assembly line balancing problem (ALBP).

ALs can be differentiated according to the assembled product variety, such as single-model, mixed-model and multi-model, and according to their layout, such as straight, U-shaped, two-sided, and parallel, etc. In parallel assembly lines (PAL), multiple ALs are positioned close and parallel to each other. When a single line is not sufficient to satisfy the demand, PAL is commonly used to increase capacity (Gökçen et al. 2006). The advantages of PAL over single assembly lines are i) improved productivity (fewer station allocations and reduction of idle time due to the use of common stations), ii) reduced failure sensitivity (when one line malfunctions, the rest of the lines may continue production) and iii) increased system flexibility (if the demand changes, each line can have a different cycle time) (Lusa 2008).

Advanced industrial organizations focus on producing multiple model types rather than concentrating on the production of a single product in order to meet a significant portion of customers' needs (Hoseinpour et al. 2020, 2021). Mixed-model AL (MMAL) is widely utilized to meet various and changing market demands. The production characteristics of the models are similar. Therefore, setup time is not required or can be ignored between different product models. Two problems should be addressed in MMAL. These are the line balancing problem (assigning tasks to stations) and the model sequencing problem (determining the production sequence of the models). These two problems are closely related (Miltenburg 2002). The sequencing of the models impacts the efficiency of the line balance, whereas the optimality of the model sequencing impacts the line balancing results (Kim et al. 2006).

In mixed-model PAL (MPAL), each line has a set of model mixes that remain constant throughout production. Different model combinations may occur in different production cycles at a common station on two neighboring lines because of the manufacturing sequence of product models. Özcan et al. (2010) addressed this problem called the mixed-model PAL balancing and model sequencing (MPALB/S) problem. In the MPALB/S problem, line balancing and model sequencing (B/S) are dealt with simultaneously.

The integration of robots in manufacturing has seen a significant increase in recent years due to advancements in technology. These machines offer a range of benefits, including manufacturing flexibility and safety, improved product quality, enhanced precision and accuracy, and reduced dependence on skilled labor (Gao et al. 2009; Goel and Gupta 2020). However, they result in a rise in energy consumption, potentially causing harmful effects on the environment. In recent years, the rise in energy prices, disruptions in energy supply, legal regulations and climate change have highlighted the importance of energy conservation.

Manufacturers are focusing on reducing energy consumption and developing more environmentally friendly production systems. This can be accomplished by implementing energy-efficient technologies, utilizing environmentally friendly energy sources and optimizing manufacturing processes.

In ALs, assembly tasks can be done by humans, robots or a collaboration of both. The assembly line where robots are used to perform assembly operations is called a robotic assembly line (RAL) (Chutima 2022). Robots with various technical specifications and capabilities may vary in processing time for a particular task. This also means that the energy consumed can vary between robots.

This paper introduces the mixed-model parallel robotic assembly line balancing and model sequencing (MPRALB/S) problem with energy consideration as a new research topic in the ALBP literature. Mixed-model parallel RAL (MPRAL) is a system in which robots replace human operators in the MPAL system. The MPRALB/S problem is an extension of the parallel RAL balancing (PRALB) (Çil et al. 2017b; Soysal-Kurt and İşleyen 2022) and MPALB/S (Özcan et al. 2010) problems, and its main characteristics are similar to them. The energy-oriented MPRALB/S problem deals with three sub-problems: i) the assignment of tasks to stations, ii) the assignment of robots to stations, and iii) the model sequencing with the lowest energy consumption. The goal is to find the best line balance while solving these sub-problems to optimize the objectives of the problem.

The authors have conducted an extensive review of the literature and found no research paper on MPRAL. Furthermore, there are very few studies that consider the energy consumption of mixed-model RAL (MRAL). Energy costs have an important place in operating costs today, it has become necessary to consider energy consumption as well as production efficiency. This paper addresses the MPRALB/S problem with energy consideration based on these findings.

The objectives of the problem are to minimize the joint cycle time and average energy consumption of MPRAL simultaneously when the number of stations is given. The MPRALB/S problem is illustrated with a numerical example. The problem has a large solution search space due to both robot and model diversity. As diversity increases, the area to be searched also increases exponentially. The problem is complex and difficult to solve. Therefore, a method based on NSGA-II is developed to solve the problem. An improvement is applied to the NSGA-II that eliminates solutions with the same objective function values. This new proposed version is called MNSGA-II. A benchmark data set for the MPRALB/S problem does not exist, hence new data sets are generated. The algorithm's parameters are set through the Taguchi method. To assess the performance of MNSGA-II, NSGA-II (without improvement) and RSA are utilized.

In this paper, Sect. 2 gives a summary of the relevant research in the field. Section 3 outlines the problem and its assumptions and presents a mathematical formulation. Section 4 gives a numerical example to illustrate the MPRALB/S problem. Section 5 introduces the proposed MNSGA-II. Section 6 contains the results of the computational study. Finally, Sect. 7 gives conclusions and suggestions for future research.

2 Literature review

MPRAL is a mixed-model version of parallel robotic assembly lines. The PRALB problem (PRALBP) was first addressed in the literature by Çil et al. (2017b). Subsequently, Soysal-Kurt and İşleyen (2022) considered PRALBP with two objective functions. PRALBP is a new research topic and there is a growing interest in PRALBP in the literature. There is no paper in the literature that investigates the MPRAL system and MPRALB/S problem considering energy consumption.

MPRAL offers various benefits such as the simultaneous production of different product models on the same line, automated and faster operations compared to traditional assembly lines, and reduced labor and resource requirements due to common station utilization. Considering that energy expenses represent a significant cost factor for businesses, balancing MPRAL with a focus on energy consumption and achieving production with lower energy usage can support businesses in achieving their environmental sustainability goals and help them gain a competitive advantage. Considering all these factors, it is inevitable that balancing and sequencing problem of MPRAL with energy consumption will receive more attention from researchers and manufacturers in the future.

This section reviews the studies on MPAL, RAL, MRAL and energy consumption in RAL, which are the most relevant topics for this paper. Cycle time and energy consumption are represented by CT and EC, respectively.

Gökçen et al. (2006) introduced the parallel ALBP (PALBP), which focuses on simultaneously balancing multiple assembly lines using common resources. To solve the PALBP, they proposed a mathematical model and heuristic method. After this study, several research studies on PAL and their variants have been reported in the literature. The problem known as MPALB/S was introduced by Özcan et al. (2010). This problem is concerned with optimizing the efficiency of production lines and balancing the workload among stations. Chutima and Yothaboriban (2017) addressed MPAL balancing problem to optimize multiple objectives hierarchically. However, they did not consider the model sequencing problem. Kucukkoc and Zhang (2014a) addressed the mixed-model parallel two-sided AL balancing and sequencing (MPTALB/S) problem. To solve the problem, they used an agent-based ant colony optimization algorithm (ABACO). Kucukkoc and Zhang (2014b) proposed a mathematical model and an ABACO for the MPTALB/S problem. Kucukkoc and Zhang (2017) presented the mixed-model parallel U-type AL B/S problem and used a heuristic for this problem. A detailed review of the studies on PALBP is given by Aguilar et al. (2020).

The studies of Rubinovitz and Bukchin (1991) and Rubinovitz et al. (1993) are the pioneers of the robotic assembly line balancing problem (RALBP). Following these studies, many different versions and solution methods of RALBP have been developed in the literature. Levitin et al. (2006) and Gao et al. (2009) addressed the RALBP to minimize the cycle time. Aghajani et al. (2014) aimed to minimize the sum of the CTs of all models for a mixed-model two-sided RAL. Rabbani et al. (2016) investigated the multi-objective mixed-model U-type RALBP including CT, robot purchasing costs, robot setup costs and sequence dependent setup costs. Çil et al. (2017a) addressed a mixed-model straight RAL to minimize the sum of the CTs of all models. Li et al. (2018) focused on the B/S of a mixed-model straight RAL to minimize the makespan. Aslan (2023) developed a variable neighborhood search algorithm to minimize the CT for the mixed-model two-sided RALBP.

Nilakantan et al. (2015) proposed two models for straight RAL, energy-based and time-based. While the objective of the energy-based model is to minimize EC, the objective of the time-based model is to minimize CT. Li et al. (2016) studied the two-sided RAL balancing problem to minimize CT and EC. Z. Zhang et al. (2019) addressed the U-type RAL to minimize CT and EC. They presented a mathematical model and used a Pareto artificial bee colony algorithm. Soysal-Kurt and İşleyen (2022) developed a hybrid firefly algorithm to simultaneously minimize the CT and EC for the PRALBP. They also presented a mathematical model. Sun et al. (2020) developed a hybrid estimation of distribution algorithm to minimize the CT and EC in a straight RAL. Zhou and Wu (2020) focused on minimizing the number of station and EC in the straight RAL. The EC includes the energy consumption of the robots during operation and standby, as well as the energy consumption during the transportation of workpieces between stations and the changeover between tasks. Nilakantan et al. (2017) aimed to maximize line efficiency and minimize the carbon footprint caused by the EC of the robots for straight RAL. B. Zhang et al. (2021) studied the mixed-model U-type RAL B/S problem to minimize the makespan and EC. Belkharroubi and Yahyaoui (2022) addressed the mixed-model straight RAL to minimize EC. Zhang et al. (2020) studied the MMAL B/S problem to optimize EC and line efficiency. They proposed a mathematical model and cellular genetic algorithm. Ngampanich and Chutima (2022) (2022 focused on the MPAL balancing problem where tasks are performed by normal workers, disabled workers and robots. The problem has multiple objectives including the energy load variance of workers, tax savings and PM 2.5 emissions. They used the EC of the robots to calculate PM 2.5 emissions. However, they did not address the model sequencing problem. Studies dealing with energy consumption in assembly lines are illustrated in Table 1.

3 Problem description

In the MPRAL system, a set of mixed-model straight RALs are arranged in parallel. The manufacturing processes of the product models are very similar. As in PAL, there are separate stations and common stations. During a production cycle, robots at separate stations work on a single model, whereas robots at common stations can work on two different models.

MPAL utilizes the advantages of PAL and MMAL. The MPRAL system proposed in this paper combines the advantages of RAL and MPAL. Some of these advantages include i) the ability to meet different customer demands and flexibility due to the ability to produce mixed product models in lines; ii) increased productivity and lower material handling costs due to the use of common resources at common stations; and iii) improved product quality and increased productivity due to the use of robots in assembly tasks.

Table 1 Papers addressing energy consumption

References	Automation level	AL layout				Type of models on each line		Problem		Objective
		Straight	U-type	Two-sided	Parallel	Single	Mixed	Balancing	Sequencing	
Nilakantan et al. (2015)	Robotic	x				x		x		Min. EC
Li et al. (2016)	Robotic			x		x		x		Min. CT and EC
Zhang et al. (2019)	Robotic		x			x		x		Min. CT and EC
Soysal-Kurt and İşleyen (2022)	Robotic				x	x		x		Min. CT and EC
Sun et al. (2020)	Robotic	x				x		x		Min. CT and EC
Zhou and Wu (2020)	Robotic	x				x		x		Min. number of station and EC
Nilakantan et al. (2017)	Robotic	x				x		x		Max. line efficiency and Min. carbon footprint
Zhang et al. (2021)	Robotic		x				x	x	x	Min. makespan and EC
Belkharroubi and Yahyaoui (2022)	Robotic	x					x	x		Min. EC
Zhang et al. (2020)	Normal workers	x					x	x	x	Min. EC and Max. line efficiency
Ngampanich and Chutima (2022)	Normal workers, disabled workers and robots				x		x	x		Min. energy load variance of workers, Max. tax savings, Min. PM 2.5 emissions, Min. average different workload between workstations and Max. line efficiency
This paper	Robotic				x		x	x	x	Min. CT and EC

A model combination that may emerge at a common station in a MPRAL may change in subsequent production cycles. The reason for this change is the order in which the product models enter the lines. In this case, it would be appropriate to consider the model sequence when balancing the lines. This is a MPRALB/S problem where balancing and sequencing problems are addressed together. As a result, more efficient solutions can be obtained by addressing these two problems together rather than separately.

Figure 1 depicts a MPRAL system. Three product models are produced in both lines. Stations 3 and 4 are separate stations. Stations 1, 2 and 5 are common stations, and during a production cycle, the robots operate on one model from each line. The product model processed at a station in a production cycle is replaced by the product model from the previous station in the next production cycle. In this case, the combination of the models occurring at a common station on parallel adjacent lines may vary.

In this paper, the objectives of the MPRALB/S problem are to minimize the average energy consumption of the lines and joint cycle time simultaneously, considering the model sequences, when the number of stations is fixed. The reason for considering the energy consumption as an average is that different model combinations occur at stations in different production cycles due to the production of mixed product models. In this case, the amount of energy consumed in each of the production cycles may be different. Therefore, this study considers the total energy consumption as the average of the energy consumption in all production cycles.

3.1 Problem assumptions

The following are the assumptions of the MPRALB/S problem considered in this study (Kucukkoc and Zhang 2014a; Nilakantan et al. 2015; Özcan et al. 2010; Soysal-Kurt and İşleyen 2022):

- Production is carried out with two or more parallel straight RALs and begins and ends at the same time on all lines.
- The production characteristics of the products are similar. There are multiple product models in each line.

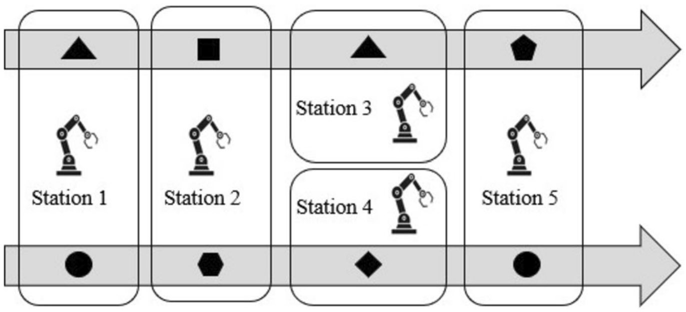


Fig. 1 The layout of MPRAL

- Each line has a set of product models, and this set remains constant throughout production.
- It is possible to work on both sides of any line.
- The model sequences are based on the concept of the minimum part set.
- The precedence relations of the product models are known. For each line, a combined precedence relations diagram (Macaskill 1972) is used. A combined precedence relations diagram $\left(P_h\right)$ is created by combining the precedence relations diagrams of all models belonging to a line.
- When assigning tasks to stations, the precedence relations must be met.
- The workload of a station cannot exceed the joint cycle time.
- Common tasks of the product models on a line must be assigned to the same station. Some tasks may have different processing times or be equal to zero for certain product models.
- Each task cannot be subdivided and must be assigned to only one station.
- Robots are capable of performing any task. The processing times of the tasks depend on the type of robot and are deterministic.
- There is only one robot at each station.
- There are no restrictions on the selection of robot types assigned to stations. A robot type can be allocated to multiple stations.
- Depending on the type of robot, the amount of energy required to complete a task varies.
- Only the energy consumed by the robots is taken into account when calculating the energy consumption of the stations.
- A robot's energy consumption consists of energy consumption in operation and standby mode.
- Material handling, loading and unloading, set-up and tool-changing times are included in the processing times of the tasks.

3.2 Notations

The following are the notations used in this paper:

h: index for lines $(h = 1, 2, \ldots, H)$

i,j: index for tasks $\left(i = 1, 2, \ldots, I_{hm}, j = 1, 2, \ldots J_{hm}\right)$

r: index for robot types $(r = 1, 2, \ldots, R)$

k: index for stations $(k = 1, 2, \ldots, K)$

m_{hm}: index for product models on line h $\left(m = 1, \ldots, M_h\right)$

H: total number of lines.

Nt: total number of tasks on all lines.

R: total number of robot types.

K: total number of stations.

M_h: total number of product models on line h.

P_h: set of precedence relations $\left(i_h, j_h\right)$, where task i is an immediate predecessor of task j in line h.

φ: index for production cycle $(\varphi = 1, \ldots, \phi)$, where $\phi = MS_{max}$

α_{k-h}^{φ}: product model produced on line h at station k in production cycle φ.

$t_{h\alpha_{h-k}^{\varphi}ir}$: processing time of task i of the model produced on line h at station k in production cycle φ with robot r.

D_{hm}: demand, over the planning horizon, for model m_{hm} produced on line h.

cd_h: greatest common divisor of product model demands $\left(D_{hm}\right)$ for line h.

MPS_h: minimum part set or model mix of line $h\left(d_h = d_{h1}, \dots d_{hM_h}\right)$

MS_h: model sequence of line h.

d_{hm}: demand for model m_{hm} in model mix of line h.

S_h: total number of products in one MPS_h for line h (the length of MS_h for one MPS_h).

S: total number of products (sum of S_h values of all lines).

$LCM\left(S_1, \dots S_H\right)$: least common multiple of S_h values.

MS_{max}: maximum number of model combinations that can appear at a common station.

TS_k: set of tasks assigned to station k.

CT: joint cycle time for all lines.

OPC_r: operation energy consumption of robot r per unit time.

SPC_r: standby energy consumption of robot r per unit time.

$ST_{k\varphi}$: workload of station k in production cycle φ.

$OEC_{k\varphi}$: operation energy consumption of station k in production cycle φ.

$SEC_{k\varphi}$: standby energy consumption of station k in production cycle φ.

$EC_{k\varphi}$: total energy consumption of station k in production cycle φ.

TEC_{φ}: total energy consumption of all lines in production cycle φ.

TEC: average energy consumption of all lines for all production cycles.

3.3 Formulation

This section gives the formulation of the MPRALB/S problem. The formulation is adapted from the studies by (Kucukkoc and Zhang 2014a; Özcan et al. 2010; Soysal-Kurt and İşleyen 2022).

A model sequence for a line is based on the minimum part set (MPS) concept (Bard et al. 1992). The minimum part set for line h $\left(MPS_h\right)$ is obtained by dividing the model demands on line h by the greatest common divisor of the demands $\left(cd_h\right)$. The vector $\left(d_h = d_{h1}, \dots d_{hM_h}\right)$ represents the model mix (minimum part set) for line h. MS_h is the model sequence of line h (Özcan et al. 2010).

$$d_{hm} = \frac{D_{hm}}{cd_h} \quad m = 1, \dots, M_h \quad h = 1, \dots, H \tag{1}$$

The length of MS_h for one MPS_h, namely, the total amount of product in one MPS_h, is calculated as:

$$S_h = \sum_{m=1}^{M_h} d_{hm} \quad h = 1, \dots, H \tag{2}$$

In MPRAL, the same model combination may not appear in all production cycles. Therefore, the system should be analyzed by dividing it into multiple production cycles ($\varphi = 1, \ldots, \phi$). By dividing the system into MS_{max} production cycles, all model combinations that may appear in MPRAL should be examined with the model mix of each line ($\phi = MS_{max}$). Each consecutive MS_{max} production cycle gives the same results. The following calculation is used to determine the maximum number of possible model combinations that may occur at a common station (Kucukkoc and Zhang 2014a):

$$MS_{max} = LCM\left(S_1, \ldots S_H\right) \quad h = 1, \ldots, H \tag{3}$$

The workload of a station may be different for each production cycle. In this case, the station time is calculated according to the model combination that appears at the relevant station in a production cycle. The workload of a station is calculated as follows:

$$ST_{k\varphi} = \sum_{i \in TS_k} t_{h\alpha^{\varphi}_{k-h}ir} \quad \forall k; \forall \varphi \tag{4}$$

After calculating the workloads of all stations for all production cycles, the maximum station time is determined and assigned as the joint cycle time of the MPRAL. The cycle time is determined as follows:

$$CT = max\left(ST_{k\varphi}\right) \quad \forall k; \forall \varphi \tag{5}$$

During a production cycle, the operation energy consumption of a station is obtained by multiplying the operation power consumption of the robot assigned to that station by the station time:

$$OEC_{k\varphi} = OPC_r . ST_{k\varphi} \quad \forall k; \forall \varphi \tag{6}$$

During a production cycle, the standby energy consumption of a station is determined by multiplying the standby power consumption of the robot assigned to that station by the standby time:

$$SEC_{k\varphi} = SPC_r . \left(CT - ST_{k\varphi}\right) \quad \forall k; \forall \varphi \tag{7}$$

During a production cycle, the total energy consumed at a station consists of the energy consumed during operation and standby:

$$EC_{k\varphi} = OEC_{k\varphi} + SEC_{k\varphi} \quad \forall k; \forall \varphi \tag{8}$$

The total energy consumption of the MPRAL during a production cycle is the sum of the energy consumption at all stations:

$$TEC_{\varphi} = \sum_{k=1}^{K} EC_{k\varphi} \quad \forall \varphi \tag{9}$$

The average energy consumption of the MPRAL is obtained by dividing the sum of the energy consumed in all production cycles by the total number of production cycles (ϕ):

$$TEC = \sum_{\varphi=1}^{\phi} TEC_{\varphi} / \phi \tag{10}$$

4 An illustrative example of the MPRALB/S problem

A small-scale example is used in this section to clearly explain the MPRALB/S problem. In this example, there are two lines, seven tasks on each line, six stations and three robot types. The MPSs for Lines 1 and 2 are $MPS_1 = (1,2)$ and $MPS_2 = (1,2)$, respectively. Since this study introduces and explains the MPRALB/S problem, the processing times and precedence relations of the product models of Line 1 are also used for Line 2. Table 2 contains a dataset for Merten's problem. The table shows the processing times (s) and precedence relations of the tasks by robots according to product models. The operation power consumptions of the robots per unit time are 0.4, 0.35 and 0.3 for Robots 1, 2 and 3, respectively. $OPC_r \times t_{h\alpha^{\varphi}_{k-h}ir}$ is used to calculate a robot's operation energy consumption. This study assumes that a robot's standby power consumption is 10% of its operation power consumption (Nilakantan et al. 2015).

A solution to Merten's problem for the MPRALB/S problem is shown in Fig. 2. In this figure, the numbers on the lines represent the indexes of the tasks, and the letters represent the product models. Stations 1 and 2 are separate stations, and the others are common stations. Table 3 shows the station workloads, station energy consumptions and model combinations that appear in the production cycles for a line balance of the mentioned problem.

Since different model combinations may appear in each production cycle in MPRAL, the number of production cycles is considered MS_{max}. $MS_{max} = LCM(3,3)$ and the maximum number of production cycles examined is three. Namely, the calculations after three production cycles will be the same as for those three production cycles.

Table 2 Immediate predecessors and processing times (s) according to product models for a Merten's problem (two product models, three robot types and seven tasks)

Tasks	Immediate predecessors	Model A			Model B		
		Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3
1	–	77	57	59	62	62	53
2	1	97	87	48	95	96	39
3	2	48	42	96	47	50	114
4	1	34	51	51	32	45	49
5	2	47	40	33	46	44	35
6	5	43	70	39	35	61	34
7	4	33	48	44	33	50	41

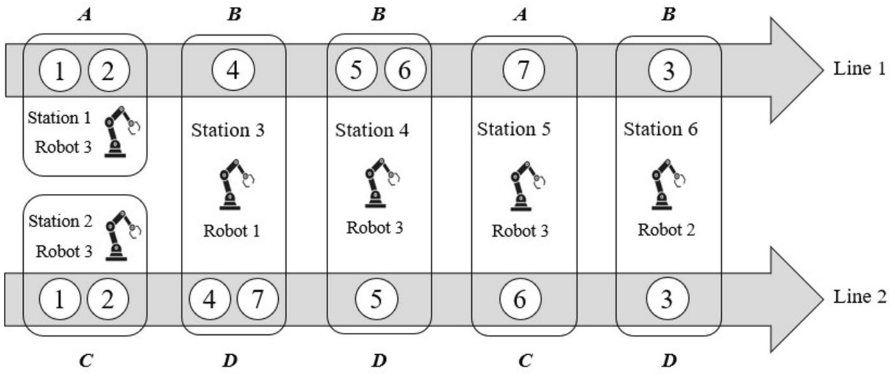


Fig. 2 A line balance for the example problem (MS_1 = BAB, MS_2 = DCD)

The model sequence of a line is placed in the table starting from the last station of the line to the first station. The first element in the model sequence of that line is placed at the last station of a line. The second element in the model sequence is placed at the previous station of the last station. The process of placing the model sequences in the table continues as mentioned until the first station of the corresponding line.

According to Table 3, the robots assigned to the stations from Station 1 to Station 6 are Robot 3, Robot 3, Robot 1, Robot 3, Robot 3 and Robot 2, respectively. Take Station 6, a common station, as an example. This station is the last station in both lines. In the first production cycle $\left(\alpha^1_{6-h}\right)$, the first elements of the model sequences of both lines appear at this station. These are model "B" for Line 1 and model "D" for Line 2. In the remaining production cycles (α^2_{6-h} and α^3_{6-h}), they appear as "AB" for Line 1 and "CD" for Line 2 according to the order in the model sequences. The tasks assigned to Station 6 are Task 3 on Line 1 and Task 3 on Line 2. Robot 2 is assigned to Station 6. During the production cycles 1, 2 and 3, the workloads of the station ($ST_{6\varphi}$) are 100, 84 and 100 s, respectively. The operation energy consumptions of the station ($OEC_{6\varphi}$) are 35, 29.4 and 35 kJ. The standby energy consumptions of the station ($SEC_{6\varphi}$) are 0.245, 0.805 and 0.245 kJ. The total energy consumptions of the station ($EC_{6\varphi}$) are 35.245, 30.205 and 35.245 kJ. The longest station time in all production cycles for all stations is the joint cycle time of the MPRAL, and therefore, the cycle time (CT) for this solution is 107 s (ST_{11}).

Table 4 shows the energy consumption of the stations and MPRAL for each production cycle. For example, Station 1 consumes 32.1 kJ, 28.05 kJ and 28.05 kJ energy during production cycles 1, 2 and 3, respectively. The average energy consumption of this station is 29.4 kJ. In addition, the energy consumed in the MPRAL is 195,555 kJ, 181,695 kJ and 185,565 kJ during production cycles 1, 2 and 3, respectively. As seen in the table, the total energy consumption of the MPRAL (TEC_{φ}) is different in all production cycles. This is because different model combinations appear in all three production cycles. Therefore, the energy consumptions for all production cycles of the MPRAL are averaged. The average energy consumption (TEC) of the MPRAL in the mentioned problem is calculated as 187.605 kJ.

Table 3 Model combinations, station workloads (s) and energy consumptions (kJ) for all production cycles

Station no	1						2						3					
Line balance	h_1	h_2	Robot 3				h_1	h_2	Robot 3				h_1	h_2	Robot 1			
	{1,2}	∅					∅	{1,2}					{4}	{4,7}				
Production cycle φ	α_{1-1}^{φ}	α_{1-2}^{φ}	$ST_{1\varphi}$	$OEC_{1\varphi}$	$SEC_{1\varphi}$	$EC_{1\varphi}$	α_{2-1}^{φ}	α_{2-2}^{φ}	$ST_{2\varphi}$	$OEC_{2\varphi}$	$SEC_{2\varphi}$	$EC_{2\varphi}$	α_{3-1}^{φ}	α_{3-2}^{φ}	$ST_{3\varphi}$	$OEC_{3\varphi}$	$SEC_{3\varphi}$	$EC_{3\varphi}$
Cycle 1	A	–	107	32.1	–	32.1	–	C	107	32.1	-	32.1	B	D	97	38.8	0.4	39.2
Cycle 2	B	–	92	27.6	0.45	28.05	–	D	92	27.6	0.45	28.05	A	C	101	40.4	0.24	40.64
Cycle 3	B	–	92	27.6	0.45	28.05	–	D	92	27.6	0.45	28.05	B	D	97	38.8	0.4	39.2
Station no	4						5						6					
Line balance	h_1	h_2	Robot 3				h_1	h_2	Robot 3				h_1	h_2	Robot 2			
	{5,6}	{5}					{7}	{6}					{3}	{3}				
Production cycle φ	α_{4-1}^{φ}	α_{4-2}^{φ}	$ST_{4\varphi}$	$OEC_{4\varphi}$	$SEC_{4\varphi}$	$EC_{4\varphi}$	α_{5-1}^{φ}	α_{5-2}^{φ}	$ST_{5\varphi}$	$OEC_{5\varphi}$	$SEC_{5\varphi}$	$EC_{5\varphi}$	α_{6-1}^{φ}	α_{6-2}^{φ}	$ST_{6\varphi}$	$OEC_{6\varphi}$	$SEC_{6\varphi}$	$EC_{6\varphi}$
Cycle 1	B	D	104	31.2	0.09	31.29	A	C	83	24.9	0.72	25.62	B	D	100	35	0.245	35.245
Cycle 2	B	D	104	31.2	0.09	31.29	B	D	75	22.5	0.96	23.46	A	C	84	29.4	0.805	30.205
Cycle 3	A	C	105	31.5	0.06	31.56	B	D	75	22.5	0.96	23.46	B	D	100	35	0.245	35.245

MS_1 = BAB, MS_2 = DCD, the number of production cycles: $LCM(3,3)$. The operation power consumptions of the robots per unit time are 0.4, 0.35 and 0.3 for Robots 1, 2 and 3, respectively

Table 4 The energy consumption at the stations and the total energy consumption of the MPRAL for all production cycles (kJ)

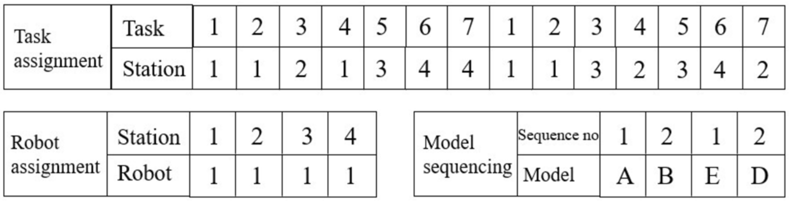
Station no	1	2	3	4	5	6	
Production cycle φ	$EC_{1\varphi}$	$EC_{2\varphi}$	$EC_{3\varphi}$	$EC_{4\varphi}$	$EC_{5\varphi}$	$EC_{6\varphi}$	TEC_{φ}
Cycle 1	32.1	32.1	39.2	31.29	25.62	35.245	195.555
Cycle 2	28.05	28.05	40.64	31.29	23.46	30.205	181.695
Cycle 3	28.05	28.05	39.2	31.56	23.46	35.245	185.565
Average consumption	29.4	29.4	39.68	31.38	24.18	33.565	187.605

5 Proposed solution approach

5.1 Solution representation

Three vectors are used to represent the solutions: task assignment vector, robot assignment vector and model sequencing vector. The task assignment vector's size is equivalent to the total number of tasks (Nt). Each position in the vector corresponds to a specific task and contains a station index. The robot assignment vector consists of K elements. Each location on the vector corresponds to a station and contains a robot type index. In the model sequencing vector, the length is S, and a product model is written into each location in the vector. A location in the vector indicates the sequence in which the product model enters the line.

The vectors representing solutions are shown in Fig. 3. For the task assignment vector, "Task" and "Station" refer to an index of a task and an index of the station where the related task is performed, respectively. According to the combined precedence relations diagram generated separately for each line, the first I_h elements are for tasks on Line 1, and the next elements are for tasks on Line 2. In the robot assignment vector, "Station" and "Robot" refer to an index of a station and an index of the robot type allocated to the corresponding station, respectively. In the model sequencing vector, the first S_h elements are for product models on Line 1, and the next elements are for tasks on Line 2. "Sequence no" refers to the sequence in which a product enters the line, and "Model" refers to the product model entering the line in the corresponding sequence number. The example in Fig. 3 is considered for two lines, two robot types, four stations and four product models. The total number of tasks on the lines is 14. In the task assignment vector, the first seven elements (tasks) are for Line 1 and the rest are for Line 2. The MPSs for Lines 1 and 2 are



Task assignment	Task	1	2	3	4	5	6	7	1	2	3	4	5	6	7
	Station	1	1	2	1	3	4	4	1	1	3	2	3	4	2

Robot assignment	Station	1	2	3	4
	Robot	1	1	1	1

Model sequencing	Sequence no	1	2	1	2
	Model	A	B	E	D

Fig. 3 A solution representation

$MPS_1 = (1, 1)$ and $MPS_2 = (1, 1)$, respectively. In the model sequencing vector, the first two elements are the model sequencing for Line 1, and the rest are for Line 2. For instance, Tasks 6 and 7 on Line 1 and Task 6 on Line 2 are performed by Robot 1 at Station 4. In the model sequencing vector, the first two elements are for Line 1 and the next two elements are for Line 2. The model sequences entering the lines are AB for Line 1 and ED for Line 2.

5.2 Modified NSGA-II

This paper employs NSGA-II, which was introduced by Deb et al. (2002), to address the MPRALB/S problem. In traditional NSGA-II, when the algorithm terminates, the population may contain many solutions with the same objective function. This situation causes the solution diversity in the population to be insufficient. The main reason for this can be explained as follows. In NSGA-II, there are parent population (PP) and offspring population (OP). The OP is formed after crossover and mutation are performed on the parents selected from the PP by methods such as tournament and roulette wheel. Then, the PP and OP are combined. The number of solutions in the combined population *(2*pop_size)* is twice the predefined population size *(pop_size)*. A new population is generated by selecting solutions from the combined population (CP) according to the fast non-dominated sorting and crowding distance method (FNSCDM) (Deb et al. 2002) until the population size *(pop_size)* is met. This population will be employed in selection, crossover and mutation operations to generate a new OP.

In the creation part of the OP, if the crossover and mutation probabilities are not met after the selection of the parents, the parents remain unchanged and are transferred to the OP. Since these solutions, which remain unchanged, appear in both the PP and OP, multiple solutions that have the same objective functions arise in the CP. In this case, duplicate solutions can be found in the final population.

This study proposes an improvement to address the issue of the NSGA-II getting trapped in local optima, which can arise from having duplicate solutions. The new proposed version is called MNSGA-II and works as follows (Algorithm 1). The initial population of the algorithm is also the PP and is created with randomly generated solutions until the population size is satisfied. Individuals with the same objective functions in the initial population are identified, one of these individuals is left in the population at random, and the others are eliminated from the population. New individuals are added to the population until the predefined population size is met. These new individuals are obtained as follows. An individual is randomly selected from the population. A new solution is created by applying mutation to this individual, and the new solution is included in the population. Thus, the initial population is created.

After this stage, offsprings are obtained by selecting parents, crossover and mutation. Then, the PP and OP are combined, and individuals with the same objective functions in the CP are found. One of these individuals is left in the CP at random and the others are removed. New individuals are added to the CP until it reaches twice the predefined population size. The procedure of creating new individuals is the same as that applied to the initial population. The FNSCDM is used to select solutions from the CP, which are then added to the new population.

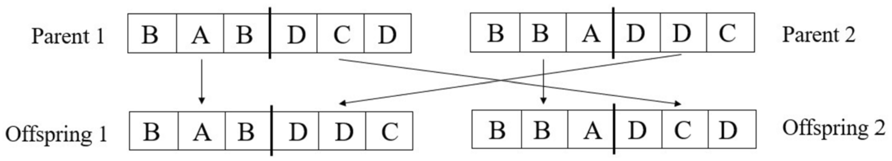


Fig. 4 Single-point crossover for the model sequencing vector

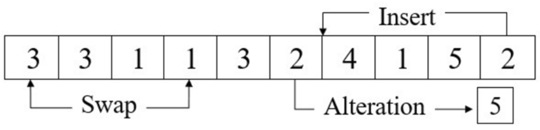


Fig. 5 Neighborhood operators for the robot assignment vector

The tournament selection method can be used for parent selection for crossover in NSGA-II. In this study, the selection of parents by tournament method is as follows. From the PP, three individuals are selected at random. One of these individuals is selected according to the FNSCDM and defined as the first parent. This selection procedure is also repeated for the selection of the second parent.

Algorithm 1 Pseudocode of MNSGA-II

```
Define objective functions $f_1(x), \dots, f_k(x)$
Initialize parameters ($pop\_size, p\_cros, p\_mut$)
Generate initial population ($P_t$)
Calculate the objective functions for each individual $x_i$ ($i = 1,2, \dots, pop\_size$)
Find the individuals with the same objective functions, leave one randomly in the population, and remove the
others from $P_t$
while ($|P_t| < pop\_size$)
    Pick a random individual $(x_j)$ from $P_t$
    Apply mutation to $x_j$ and add the new individual to $P_t$
    Calculate the objective functions of individuals in $P_t$
    Find the individuals with the same objective functions, leave one randomly in the population, and remove the
    others from $P_t$
end while
while (termination criterion is not satisfied)
    for $i = 1: (pop\_size/2)$
        Select parents from $P_t$ with tournament selection
        Perform crossover and mutation with parents and create the offspring population ($Q_t$)
    end for
    Combine $P_t$ and $Q_t$ to get the population $R_t$
    Calculate the objective functions of individuals in $R_t$
    Find the individuals with the same objective functions, leave one randomly in the population, and remove the
    others from $R_t$
    while ($|R_t| < (2 * pop\_size)$)
        Pick a random individual $(x_j)$ from $R_t$
        Apply mutation to $x_j$ and add the new individual to $R_t$
        Calculate the objective functions of individuals in $R_t$
        Find the individuals with the same objective functions, leave one randomly in the population, and remove
        the others from $R_t$
    end while
    Create $P_{t+1}$ by selecting individuals from $R_t$ according to the FNSCDM.
end while
```

5.3 Crossover and mutation operations

The crossover and mutation operations are performed if the predetermined crossover and mutation probabilities are satisfied. In this study, the crossover operation is applied for all three vectors. Single-point crossover is applied for crossover. The crossover operator for the model sequencing vector is performed as follows (Fig. 4). Parent 1's model sequence on Line 1 is Offspring 1's model sequence on Line 1. Parent 2's model sequence on Line 2 is Offspring 1's model sequence on Line 1. Thus, Offspring 1 is obtained. For Offspring 2, Parent 2's model sequence on Line 1 is Offspring 2's model sequence on Line 1. Parent 1's model sequence on Line 2 is Offspring 2's model sequence on Line 2.

A mutation is applied to the offspring after the crossover. For mutation, three neighborhood operators are applied. These are swap, insert and alteration operators. The following is the framework of the neighborhood search. The algorithm randomly selects one of the three vectors to search for neighborhoods. Then it chooses Line 1 or Line 2 at random. By randomly applying one of the neighborhood operators, it generates a new neighbor solution. The alteration operator can be applied to the task and robot assignment vectors. The swap and insertion operators can be applied to all three vectors.

The neighborhood operators can be described as follows (Fig. 5). By using the swap operator, the values at two randomly selected locations in the vector are exchanged. With the insert operator, a location is selected at random, and the value at that place is inserted into a different location. By using the alteration operator, a location is selected at random and a different value is assigned to it.

In this study, a repair mechanism is used to make newly generated solutions feasible if they do not meet the precedence constraints of the tasks. If a task is assigned to a station after its successor task, their station indices are swapped (Nilakantan et al. 2017). If the vector does not contain all station indices, the missing indices are inserted into the vector at random until all are included, ensuring that every station has at least one assigned task (Soysal-Kurt and İşleyen 2022).

6 Computational study

A computational study is conducted in this section to solve the multi-objective MPRALB/S problem. The study includes a comparison of the proposed MNSGA-II with NSGA-II and RSA (Li et al. 2016). NSGA-II is implemented to test the effectiveness of the proposed improvement in MNSGA-II. Also, RSA is used for comparison because it is effective at solving multi-objective problems. The RSA's neighborhood search mechanism is performed similarly to the MNSGA-II's mutation procedure.

New datasets are generated for the MPRALB/S problem, as there is no existing dataset in the literature. The processing times of Line 1 in the paper by Çil et al. (2017b) are taken as a basis to generate the processing times of the product models. The processing times in the relevant paper are used as the processing times for

Table 5 Problem instances examined in the paper

Problem/Tasks (Line 1-Line 2)	The number of robot types	The number of stations	Minimum part set $(MPS_1) - (MPS_2)$
Merten/(7–7)	2	4	(1,1)–(1,1)
	3	6	(1,2)–(1,2)
	4	8	(1,2)–(2,1)
Jaeschke/(9–9)	2	4	(1,1)–(1,1)
	3	6	(1,2)–(1,2)
	4	8	(1,2)–(2,1)
Roszieg/(25–25)	3	6	(1,1)–(1,1)
	4	8	(1,2)–(1,2)
	6	12	(1,2)–(2,1)
Gunther/(35–35)	5	10	(1,1)–(1,1)
	7	14	(1,2)–(1,2)
	12	24	(1,2)–(2,1)
Hahn/(53–53)	5	10	(1,1)–(1,1)
	10	20	(1,2)–(1,2)
	14	28	(1,2)–(2,1)
Tonge/(70–70)	7	14	(1,1)–(1,1)
	10	20	(1,2)–(1,2)
	14	28	(1,2)–(2,1)
Lutz3/(89–89)	8	16	(1,1)–(1,1)
	12	24	(1,2)–(1,2)
	16	32	(1,2)–(2,1)

model "A" on Line 1. The processing times of model "B" on Line 1 are generated by multiplying the processing time of model "A" for robot R_i by random values in the range $[0.8 \times t_{ir}, 1.2 \times t_{ir}]$. The data set generated for Line 1 is also used for Line 2. The power consumption of the robots per unit time is taken from the studies by (Nilakantan et al. 2015; Soysal-Kurt and İşleyen 2022). The precedence relations between tasks can be found on the website http://assembly-line-balancing.de/.

Table 5 gives the problem instances used in the study. For instance, the expression Hahn/53–53 in the table indicates that there are 53 tasks on Line 1 and 53 tasks on Line 2 in the Hahn problem. The number of stations is certain and fixed for each problem. The minimum part sets for the lines are $(MPS_1) - (MPS_2)$. Algorithm parameters are set with Taguchi method on Minitab 19. All the algorithms are encoded in Python and run on Anaconda3-Spyder with an Intel(R) Core(TM) i5-6400 CPU, 2.70 GHz, and 8 GB RAM.

6.1 Parameter setting with Taguchi method

The Taguchi method is useful for determining the best combination of parameter levels when there is multiple parameter and levels. It can be applied when the number of full factorial tests for determining parameter levels is high and can reach a conclusion with fewer tests by employing orthogonal arrays. The test results are evaluated with a signal-to-noise (SN) ratio.

Hypervolume ratio (HVR) is a significant measure that assesses the solution diversity and proximity to the Pareto front. Therefore, the response of the experiments is the value of (1-HVR). The objectives of the MPRALB/S problem have different measurement units. To calculate the HVR more consistently, the Pareto solutions are normalized according to the objective functions. The true Pareto front, which is the set of optimal solutions for a given problem, is required to calculate the HVR. The true Pareto front, which cannot be determined in this problem, is composed of the non-dominated solutions obtained after running all the experiments.

For parameter setting, Hahn problem with ten stations is used. The termination criterion for the algorithms is $Nt \times Nt \times 40$ ms CPU time, and all experiments are repeated five times. MNSGA-II has three parameters, each with three levels. Thus, the orthogonal array L9 is used, and the design of the experiments is presented in Table 6.

The parameters and levels of the MNSGA-II are as follows:

- Population size *(pop_size)*: 30, 60, 90
- Probability of crossover *(p_cros)*: 0.7, 0.8, 0.9
- Probability of mutation *(p_mut)*: 0.3, 0.4, 0.5.

Figure 6 and Table 7 show the parameter setting results of the MNSGA-II. Table 7 is the response table for the SN ratios to determine the best levels of the parameters. "Delta" and "Rank" indicate the effect of the parameters on the response. The largest Delta value gives the most important parameter. The ranks indicate the order of effect of the parameters. Thus, the most important parameter is the population size (*pop_size*) with the largest Delta value, followed by the probability of mutation (*p_mut*) and the probability of crossover (*p_cros*). In the main effects plot (Fig. 6), the highest average SN ratio gives the best level for each parameter. Therefore, the best parameter levels for solving the problem are *pop_size* = 30, *p_cros* = 0.9 and *p_mut* = 0.3. The parameter values for MNSGA-II are also used for NSGA-II.

The same procedure as for the MNSGA-II is applied for the parameter setting of the RSA. The followings are the parameters and levels:

- Initial temperature (*T0*): 0.5, 0.75, 1.0
- Cooling rate (*alpha*): 0.90, 0.95, 0.98
- Number of solutions searched in each temperature (*N*): 10, 20, 30
- Number of moves before restart (*NR*): 50, 100, 150.

The best parameter levels for the RSA are *T0* = 0.75, *alpha* = 0.95, *N* = 30 and *NR* = 50.

Table 6 Orthogonal array for parameter setting

Experiment	Levels of parameters		
	Population size (*pop_size*)	Probability of crossover (*p_cros*)	Probability of mutation (*p_mut*)
1	30	0.7	0.3
2	30	0.8	0.4
3	30	0.9	0.5
4	60	0.7	0.4
5	60	0.8	0.5
6	60	0.9	0.3
7	90	0.7	0.5
8	90	0.8	0.3
9	90	0.9	0.4

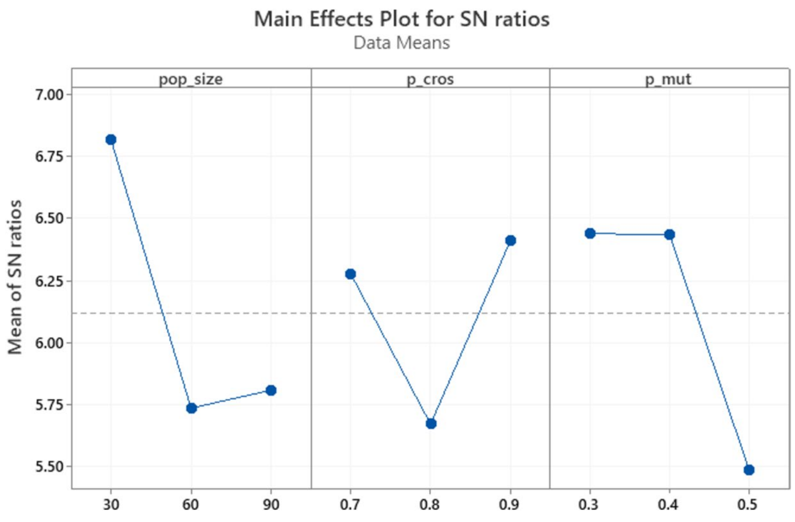


Fig. 6 SN ratios mean effect plot

Table 7 Responses for SN ratios

Level	*pop_size*	*p_cros*	*p_mut*
1	6.814	6.278	6.439
2	5.738	5.673	6.432
3	5.810	6.410	5.490
Delta	1.076	0.737	0.948
Rank	1	3	2

6.2 Performance measures

In this paper, the performance of the algorithms is evaluated through three widespread evaluation metrics: hypervolume (HV) ratio, the ratio of non-dominated solutions (RP) and generational distance (GD). The followings are descriptions of the metrics:

Hypervolume ratio. It is calculated by dividing the HV of a Pareto frontier (S) by the HV of the true Pareto front (P) and is expressed as follows (Nilakantan et al. 2017):

$$HVR = \frac{volume\left(\bigcup_{i=1}^{size(S)} v_i\right)}{volume\left(\bigcup_{j=1}^{size(P)} v_j\right)} \tag{11}$$

v_i and v_j refer to hypercubes. The worst values for all objectives are represented by a reference point W. It is used to determine a Pareto set's volume. The diagonal corners of a hypercube are on the objective vectors of the Pareto frontier and W. The higher the HVR of a Pareto set, the closer it is to the true Pareto front.

The ratio of non-dominated solutions. It is the ratio of the number of solutions non-dominated by the other non-dominated solutions in the union of sets to the number of non-dominated solutions produced by the algorithm (Soysal-Kurt and İşleyen 2022; Zhang et al. 2019).

$$RP\left(S_i\right) = \left|S_i - \left\{x \in S_i \middle| \exists y \in S : y \succ x\right\}\right| / S_i \tag{12}$$

where $y \succ x$ means y dominates x, $\left|S_i\right|$ is the number of solutions in the obtained Pareto set S_i by the algorithm, and S is the union of all sets. It is the number of non-dominated solutions produced by the algorithm in the union of sets divided by the number of non-dominated solutions produced by the algorithm. A higher value of this metric indicates a better set of solutions.

Generational distance. It is used to measure the distance between a Pareto set and the true Pareto front and is calculated as the average Euclidean distance between the solutions in the Pareto set and their nearest point on the true Pareto front. A lower GD value indicates a better solution set. GD is described as (Van Veldhuizen and Lamont 2000):

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{13}$$

where n is the number of solutions in the Pareto set obtained by an algorithm, and d_i is the Euclidean distance between solution i and its nearest point on the true Pareto front.

The non-dominated solutions obtained by the algorithms are normalized to compute the metrics consistently. Furthermore, the true Pareto fronts for each test problem consist of non-dominated solutions that occur after the union of the Pareto sets obtained by all algorithms.

6.3 Comparison of algorithms

This section focuses on evaluating the MNSGA-II, NSGA-II and RSA in terms of HVR, GD and RP metrics. For each problem instance, the experiments are run ten times due to the randomness, and the termination criterion of the algorithms is $Nt \times Nt \times 100$ ms CPU time.

Table 8 shows the results of the performance metrics for each problem instance. Evaluation of the findings reveals that in terms of the HVR metric, MNSGA-II achieves the best values among all algorithms in 12 cases, RSA in 10 cases and NSGA-II in 3 cases. In terms of the RP metric, both MNSGA-II and RSA obtain the best values among all algorithms in 12 cases and NSGA-II in 3 cases. Regarding the GD metric, MNSGA-II achieves the best values in 13 cases, RSA in 11 cases and NSGA-II in 2 cases. Additionally, it is observed that in multiple problem instances, certain algorithms achieve identical metric values.

When comparing the performances of MNSGA-II and NSGA-II to assess the effectiveness of the proposed improvement in MNSGA-II, it is clear that MNSGA-II outperforms NSGA-II across all performance metrics. MNSGA-II shows superior performance in approximately 81% of the cases for the HVR, 71% of the cases for the RP and 76% of the cases for the GD.

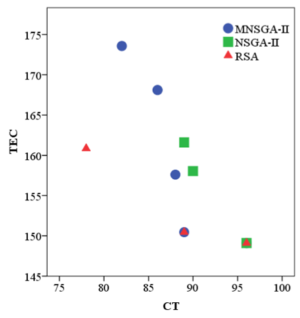
Figure 7 shows the non-dominated solutions obtained from the algorithms for six problem instances with different sizes. All algorithms can obtain many solutions within ten runs, but figures include the non-dominated solutions from all runs. The non-dominated solutions are indicated by blue circles for MNSGA-II, green squares for NSGA-II and red triangles for RSA. The computational results and Fig. 7 clearly demonstrate the superiority of the proposed MNSGA-II over NSGA-II. Moreover, it can be inferred that MNSGA-II and RSA show comparable performance in solving the problem instances.

To explain the trade-off between the objective functions from a managerial perspective, consider the non-dominated solutions generated by MNSGA-II in the "Gunther with 10 stations" problem illustrated in Fig. 7c. The Pareto front of the MNSGA-II has six non-dominated solutions, labeled as "p1" to "p6". The "p1" solution has a cycle time of 340 s and an average energy consumption of 1829.16 kJ, whereas "p6" has a cycle time of 357 s and an average energy consumption of 1684.65 kJ. By moving from "p1" to "p6" on the Pareto front, there is an opportunity for 7.9% energy savings with a longer cycle time of 5%. On the other hand, some positions on the front offer energy savings at a minimal cost to cycle time, such as moving from point "p4" to point "p5" which results in a 26.36 kJ energy savings while only extending the cycle time by 2 s. Furthermore, a shift from solution "p2" to "p3" on the Pareto front results in a 55.69 kJ reduction in average energy consumption, with an increase of two seconds in the cycle time. When this approach is applied to large-scale problems, significant energy savings can be achieved, and it

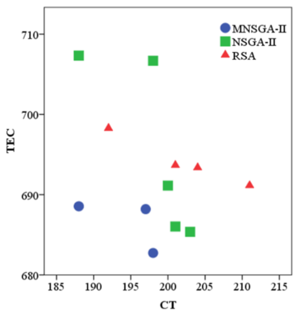
Table 8 Comparison of the algorithms by performance metrics

Problem/Tasks (Line 1-Line 2)	St	HVR			RP			GD		
		MNSGA-II	NSGA-II	RSA	MNSGA-II	NSGA-II	RSA	MNSGA-II	NSGA-II	RSA
Merten/7–7	4	*1.000*	0.008	*1.000*	*1.000*	0.000	*1.000*	*0.000*	1.189	*0.000*
	6	*1.000*	0.114	*1.000*	*1.000*	0.000	*1.000*	*0.000*	0.737	*0.000*
	8	0.683	0.430	*0.992*	0.600	0.333	*1.000*	0.201	0.178	*0.000*
Jaeschke/9–9	4	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*1.000*	*0.000*	*0.000*	*0.000*
	6	0.650	0.705	*1.000*	0.250	*0.750*	*0.750*	0.182	0.126	*0.015*
	8	0.647	*1.000*	0.300	0.200	*1.000*	0.250	0.205	*0.000*	0.288
Roszieg/25–25	6	0.677	0.562	*1.000*	0.500	0.111	*0.750*	0.241	0.214	*0.021*
	8	*1.000*	0.842	0.761	*1.000*	0.000	0.000	*0.000*	0.277	0.378
	12	*1.000*	0.556	0.494	*1.000*	0.000	0.000	*0.000*	0.291	0.327
Gunther/35–35	10	*0.923*	0.779	0.900	0.167	0.333	*0.800*	*0.132*	0.206	0.146
	14	*0.996*	0.670	0.850	*1.000*	0.000	0.286	*0.000*	0.195	0.127
	24	*1.000*	0.078	0.448	*1.000*	0.000	0.000	*0.000*	1.006	0.622
Hahn/53–53	10	0.802	0.576	*1.000*	0.000	0.000	*1.000*	0.125	0.190	*0.000*
	20	0.702	0.080	*0.999*	0.333	0.000	*1.000*	0.333	0.600	*0.000*
	28	0.816	*0.961*	0.219	*1.000*	0.500	0.000	*0.000*	0.138	0.399
Tonge/70–70	14	0.676	0.072	*1.000*	0.000	0.000	*1.000*	0.290	0.502	*0.000*
	20	*0.980*	0.110	0.383	*1.000*	0.000	*1.000*	*0.000*	0.466	*0.000*
	28	*0.995*	0.147	0.356	*1.000*	0.000	0.200	*0.000*	0.457	0.239
Lutz3/89–89	16	0.657	0.388	*0.972*	0.091	0.000	*1.000*	0.158	0.331	*0.000*
	24	*1.000*	0.480	0.661	*1.000*	0.000	0.000	*0.000*	0.278	0.275
	32	*0.979*	0.224	0.230	*1.000*	0.000	0.500	*0.000*	0.542	0.250

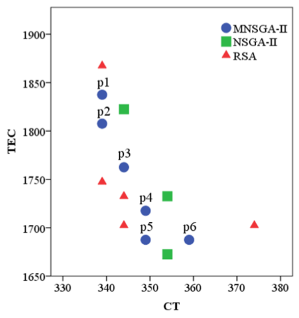
The values that are better for the relevant metric are shown in italics



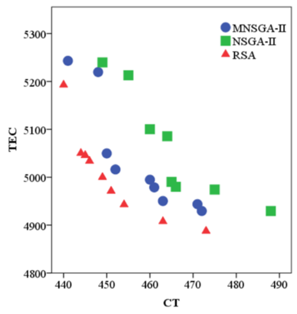
(a) Merten with 8 stations



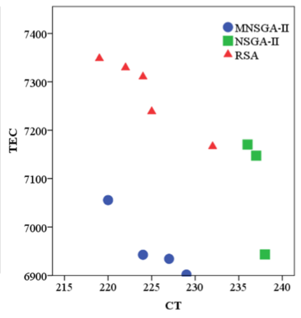
(b) Roszieg with 12 stations



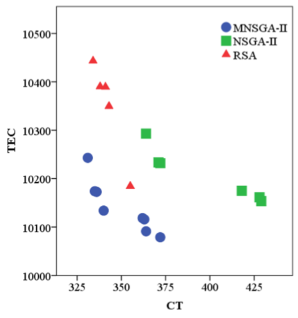
(c) Gunther with 10 stations



(d) Hahn with 10 stations



(e) Tonge with 28 stations



(f) Lutz3 with 24 stations

Fig. 7 Pareto fronts of the MNSGA-II, NSGA-II and RSA

can guide managers who want to reduce the negative environmental impacts of production while determining the most appropriate production plan.

In addition, MNSGA-II, NSGA-II and RSA are statistically compared using performance measures. An initial ANOVA test is utilized, and it is determined that the performance metric values are not normally distributed. Therefore, the non-parametric Friedman rank-based test is conducted to assess whether there is a statistically significant difference between the algorithms. Since higher values are better for HVR and RP, if the rank value for one algorithm is higher than the rank values of the other algorithms, this algorithm can be considered to perform better than the others. For GD, the opposite is true as lower value is better. Figure 8 shows the Friedman ranks of HVR, RP and GD. For the HVR metric (Fig. 8a), the Friedman ranks indicate that MNSGA-II performs the best, followed by RSA and NSGA-II. In the case of RP (Fig. 8b), the ranks remain RSA, MNSGA-II and NSGA-II. However, it's important to note that the rank values for MNSGA-II and RSA are nearly identical, suggesting their comparable performance. Lastly, when considering the GD metric (Fig. 8c), the rankings are MNSGA-II, RSA and NSGA-II. These ranks provide valuable insights into the relative performance of these algorithms across different evaluation criteria.

According to Friedman test statistics, *p* values are 0.001, 0.003 and 0.003 for HVR, RP and GD, respectively. The results at a 95% confidence level indicate that

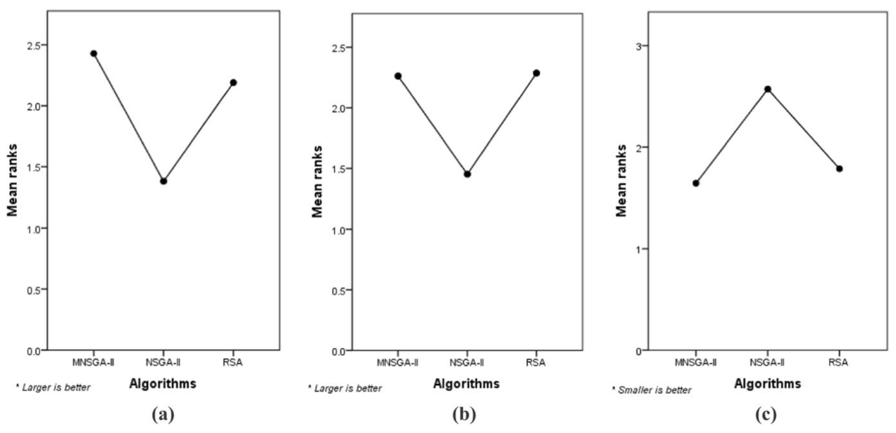


Fig. 8 The Friedman ranks of HVR (**a**), RP (**b**) and GD (**c**)

the performance of at least one algorithm is different from the others. The Wilcoxon signed rank test is applied to determine which algorithms' performances are different. The *p*-values for the Wilcoxon test are given in Table 9. A *p* value less than 0.05 means that there is a significant difference.

According to the statistical test results, the following conclusions can be drawn. i) The performances of NSGA-II and MNSGA-II are different and MNSGA-II outperforms NSGA-II. ii) The performances of RSA and NSGA-II are different and RSA outperforms NSGA-II. iii) There is no statistically significant difference between the performances of MNSGA-II and RSA.

7 Conclusion

Robots have been widely used in the industrial sector in recent years. While they increase production efficiency significantly, they also increase energy consumption and carbon emissions. In manufacturing, robots are commonly utilized on assembly lines. Consideration of energy consumption when balancing a robotic assembly line can result in significant energy savings. In MPRAL, where multiple product models are produced in a mixed sequence, finding an optimal line balance and model sequence is quite difficult. As a result, this paper aims to minimize cycle time and energy consumption in the MPRALB/S problem, and it is the first attempt to simultaneously address balancing and sequencing problems in MPRAL.

The MPRALB/S problem aiming to minimize the cycle time and energy consumption is illustrated with a numerical example. A MNSGA-II is developed to solve the problem. This algorithm eliminates solutions with the same objective function values in the population. In order to determine the parameter values of the algorithm, the Taguchi method is employed. The algorithm's performance is then compared to NSGA-II and RSA. The results of the computational study show that

Table 9 Statistics for the Wilcoxon signed ranks test

	p-values		
Performance metrics	NSGA-II–MNSGA-II	RSA–NSGA-II	RSA–MNSGA-II
HVR	0.001	0.010	0.122
RP	0.002	0.006	0.474
GD	0.001	0.005	0.309

MNSGA-II is superior to NSGA-II. Furthermore, MNSGA-II and RSA produce solutions with similar levels of diversity and proximity to the true Pareto front.

Line balancing and model sequencing are two closely related problems for a MRAL. The performance of both problems is affected by each other's results. The assignment of robots to stations by considering their energy consumption complicates the process of determining the best solution for the MPRALB/S problem. There are some benefits of simultaneously balancing and model sequencing of MPRAL to minimize cycle time and energy consumption, such as increased line efficiency and reduced energy consumption. Since processing times may vary from one model to another, assigning suitable tasks to stations can reduce station idle times, resulting in increased productivity. Therefore, as the energy consumed in the MPRAL will also decrease, a sustainable line is obtained and an economic improvement is provided for the company.

As the addressed problem introduces a new field of research to the assembly line literature, it provides opportunities for researchers to further develop this field. Any robotic assembly line balancing problem is known to be complex. It becomes more complicated when the model sequencing and balancing problems are addressed simultaneously. It takes more time to find an effective solution. In future studies, solution approaches such as mathematical models, heuristic and metaheuristic methods that find effective solutions in a shorter time for more complex model mixtures can be developed.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10696-024-09533-1.

Acknowledgements This paper is extracted from the doctoral thesis of the author Halenur Soysal-Kurt at Gazi University.

Author contributions HSK: Writing—Original Draft, Writing—review & editing, Software, Data curation; SKİ: Methodology, Supervision; HG: Methodology, Supervision.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). No funds, grants, or other support was received.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Aghajani M, Ghodsi R, Javadi B (2014) Balancing of robotic mixed-model two-sided assembly line with robot setup times. Int J Adv Manuf Technol 74(5–8):1005–1016. https://doi.org/10.1007/s00170-014-5945-x

Aguilar H, García-Villoria A, Pastor R (2020) A survey of the parallel assembly lines balancing problem. Comput Oper Res 124:105061. https://doi.org/10.1016/j.cor.2020.105061

Aslan Ş (2023) Mathematical model and a variable neighborhood search algorithm for mixed-model robotic two-sided assembly line balancing problems with sequence-dependent setup times. Optim Eng 24(2):989–1016. https://doi.org/10.1007/s11081-022-09718-3

Bard JF, Dar-El E, Shtub A (1992) An analytic framework for sequencing mixed model assembly lines. Int J Prod Res 30(1):35–48. https://doi.org/10.1080/00207549208942876

Belkharroubi L, Yahyaoui K (2022) Solving the energy-efficient robotic mixed-model assembly line balancing problem using a memory-based cuckoo search algorithm. Eng Appl Artif Intell 114:105112. https://doi.org/10.1016/j.engappai.2022.105112

Chutima P (2022) A comprehensive review of robotic assembly line balancing problem. J Intell Manuf 33(1):1–34. https://doi.org/10.1007/s10845-020-01641-7

Chutima P, Yothaboriban N (2017) Multi-objective mixed-model parallel assembly line balancing with a fuzzy adaptive biogeography-based algorithm. Int J Ind Syst Eng 26(1):90–132. https://doi.org/10.1504/IJISE.2017.083182

Çil ZA, Mete S, Ağpak K (2017a) Analysis of the type II robotic mixed-model assembly line balancing problem. Eng Optim 49(6):990–1009. https://doi.org/10.1080/0305215X.2016.1230208

Çil ZA, Mete S, Özceylan E, Ağpak K (2017b) A beam search approach for solving type II robotic parallel assembly line balancing problem. Appl Soft Comput 61:129–138. https://doi.org/10.1016/j.asoc.2017.07.062

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

Gao J, Sun L, Wang L, Gen M (2009) An efficient approach for type II robotic assembly line balancing problems. Comput Ind Eng 56(3):1065–1080. https://doi.org/10.1016/j.cie.2008.09.027

Goel R, Gupta P (2020) Robotics and Industry 4.0. In: Nayyar A, Kumar A (eds) A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development. Springer International Publishing, pp 157–169. https://doi.org/10.1007/978-3-030-14544-6_9

Gökçen H, Ağpak K, Benzer R (2006) Balancing of parallel assembly lines. Int J Prod Econ 103(2):600–609. https://doi.org/10.1016/j.ijpe.2005.12.001

Hoseinpour Z, Kheirkhah AS, Fattahi P, Taghipour M (2020) The problem solving of bi-objective hybrid production with the possibility of production outsourcing through meta-heuristic algorithms. Management 4(2):1–17. https://doi.org/10.31058/j.mana.2021.42001

Hoseinpour Z, Taghipour M, Beigi JH, Mahboobi M (2021) The problem solving of bi-objective hybrid production with the possibility of production outsourcing through imperialist algorithm, NSGA-II, GAPSO hybrid algorithms. Turk J Comput Math Edu 12(13):8090–8111

Kim YK, Kim JY, Kim Y (2006) An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines. Eur J Oper Res 168(3):838–852. https://doi.org/10.1016/j.ejor.2004.07.032

Kucukkoc I, Zhang DZ (2014a) Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. Int J Prod Res 52(12):3665–3687. https://doi.org/10.1080/00207543.2013.879618

Kucukkoc I, Zhang DZ (2014b) Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. Int J Prod Econ 158:314–333. https://doi.org/10.1016/j.ijpe.2014.08.010

Kucukkoc I, Zhang DZ (2017) Balancing of mixed-model parallel U-shaped assembly lines considering model sequences. Int J Prod Res 55(20):5958–5975. https://doi.org/10.1080/00207543.2017.1312586

Levitin G, Rubinovitz J, Shnits B (2006) A genetic algorithm for robotic assembly line balancing. Eur J Oper Res 168(3):811–825. https://doi.org/10.1016/j.ejor.2004.07.030

Li Z, Tang Q, Zhang L (2016) Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. J Clean Prod 135:508–522. https://doi.org/10.1016/j.jclepro.2016.06.131

Li Z, Janardhanan MN, Tang Q, Nielsen P (2018) Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line. Eng Optim 50(5):877–893. https://doi.org/10.1080/0305215X.2017.1351963

Lusa A (2008) A survey of the literature on the multiple or parallel assembly line balancing problem. Eur J Ind Eng 2(1):50–72. https://doi.org/10.1504/EJIE.2008.016329

Macaskill JLC (1972) Production-line balances for mixed-model lines. Manag Sci 19(4-part-1):423–434. https://doi.org/10.1287/mnsc.19.4.423

Miltenburg J (2002) Balancing and scheduling mixed-model U-shaped production lines. Int J Flex Manuf Syst 14:119–151. https://doi.org/10.1023/A:1014434117888

Ngampanich S, Chutima P (2022) Many-objective mixed-model parallel assembly line balancing utilizing normal workers, disabled workers, and robots. In: 2022 4th International Conference on Management Science and Industrial Engineering (MSIE), pp 311–317. https://doi.org/10.1145/3535782.3535823

Nilakantan JM, Huang GQ, Ponnambalam SG (2015) An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. J Clean Prod 90:311–325. https://doi.org/10.1016/j.jclepro.2014.11.041

Nilakantan JM, Li Z, Tang Q, Nielsen P (2017) Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems. J Clean Prod 156:124–136. https://doi.org/10.1016/j.jclepro.2017.04.032

Özcan U, Çerçioğlu H, Gökçen H, Toklu B (2010) Balancing and sequencing of parallel mixed-model assembly lines. Int J Prod Res 48(17):5089–5113. https://doi.org/10.1080/00207540903055735

Rabbani M, Mousavi Z, Farrokhi-Asl H (2016) Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem. J Ind Prod Eng 33(7):472–484. https://doi.org/10.1080/21681015.2015.1126656

Rubinovitz J, Bukchin J, Lenz E (1993) RALB–a heuristic algorithm for design and balancing of robotic assembly lines. CIRP Ann 42(1):497–500. https://doi.org/10.1016/S0007-8506(07)62494-9

Rubinovitz J, Bukchin J (1991) Design and balancing of robotic assembly lines. In: Proceedings of the Fourth World Conference on Robotics Research, Pittsburgh, PA.

Soysal-Kurt H, İşleyen SK (2022) Multi-objective optimization of cycle time and energy consumption in parallel robotic assembly lines using a discrete firefly algorithm. Eng Comput 39(6):2424–2448. https://doi.org/10.1108/EC-12-2020-0747

Sun B, Wang L, Peng Z (2020) Bound-guided hybrid estimation of distribution algorithm for energy-efficient robotic assembly line balancing. Comput Ind Eng 146:106604. https://doi.org/10.1016/j.cie.2020.106604

Van Veldhuizen DA, Lamont GB (2000) On measuring multiobjective evolutionary algorithm performance. İn: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), 1, pp 204–211. Doi: https://doi.org/10.1109/CEC.2000.870296

Zhang Z, Tang Q, Li Z, Zhang L (2019) Modelling and optimisation of energy-efficient U-shaped robotic assembly line balancing problems. Int J Prod Res 57(17):5520–5537. https://doi.org/10.1080/00207543.2018.1530479

Zhang B, Xu L, Zhang J (2020) A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line. J Clean Prod 244:118845. https://doi.org/10.1016/j.jclepro.2019.118845

Zhang B, Xu L, Zhang J (2021) Balancing and sequencing problem of mixed-model U-shaped robotic assembly line: mathematical model and dragonfly algorithm based approach. Appl Soft Comput 98:106739. https://doi.org/10.1016/j.asoc.2020.106739

Zhou B-H, Wu Q (2020) Decomposition-based bi-objective optimization for sustainable robotic assembly line balancing problems. J Manuf Syst 55:30–43. https://doi.org/10.1016/j.jmsy.2020.02.005

Halenur Soysal Kurt graduated from Yıldız Technical University, Department of Industrial Engineering in 2011. She received the M.Sc. degrees in Industrial Engineering from Sakarya University in 2014 and in Management Information Systems from Osmaniye Korkut Ata University in 2017. She received her Ph.D. degree in Management Information Systems from Gazi University in 2023. Her research areas mainly focus on optimization problems, robotic assembly line balancing, decision support systems and metaheuristic algorithms.

Selçuk Kürşat İşleyen graduated from the Faculty of Engineering at Gazi University. He received M.Sc. and Ph.D. degrees in Industrial Engineering from Gazi University in 2004 and 2008, respectively. His research focus is on logistics and assembly line balancing problems.

Hadi Gökçen graduated from Industrial Engineering Department in Gazi University. In the years of 1989 and 1994, he received M.Sc. and Ph.D. degrees in Industrial Engineering from Gazi University. Dr. Gökçen received the title of Associate Professor from manufacturing and service systems in 1998 and title of Professor in 2004. His research interests include optimization theory and methods, production planning and control, assembly line balancing, decision support systems and logistic management. Dr. Gökçen has over a hundred national and international publications and has an H-index of 22.