International Journal of Industrial Engineering Computations 16 (2025) 69–92

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Multi-objective mixed-model assembly line balancing with hierarchical worker assignment: A case study of gear reducer manufacturing operations

He-Yau Kang[a], Amy H. I. Lee[b*] and Yi-Xuan Su[a]

[a]*Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung, Taiwan*
[b]*Department of Industrial Management, Chung Hua University, Hsinchu, Taiwan*

C H R O N I C L E

Article history:
Received September 2 2024
Received in Revised Format September 29 2024
Accepted October 28 2024
Available online October 28 2024

Keywords:
Mixed-model assembly line balancing problem (MALBP)
Hierarchical workforce
Mixed integer programming (MIP)
Multi-objective genetic algorithm (MOGA)
Non-dominated sorting genetic algorithm II (NSGA-II)

A B S T R A C T

Assembly lines, generally speaking, can reduce production costs, shorten cycle times, and achieve higher quality levels. Since the current market is characterized by increasing product variability, mixed-model assembly lines, in which similar product models can be assembled simultaneously, are more suitable to respond to varied market demands than traditional single-model assembly lines. In addition, in an assembly line, tasks often differ in processing requirements, and workers may have different qualification levels. This study, therefore, aims to construct models for the multi-objective mixed-model assembly line balancing problem with hierarchical worker assignment (MO-MALBP-HW). The goal is to generate a suitable plan for a mixed-model assembly line balancing problem considering the constraint of a hierarchical workforce, the cost of a hierarchical workforce, and production cycle time. When the problem is simple, it can be solved by a mixed integer programming (MIP) model. When the problem becomes complex, it can be solved by a multi-objective genetic algorithm (MOGA) and a non-dominated sorting genetic algorithm II (NSGA-II) to obtain a near-optimal solution. The implementation of this model can effectively manage the multi-objective mixed-model assembly line balancing plan, thereby improving plant efficiency and reducing cost.

1. Introduction

Assembly line is an important part of a mass production manufacturing system. The assembly line balancing problem (ALBP) refers to finding the optimal assignment of tasks into a set of workstations to achieve maximum production efficiency (Li et al., 2023). Among the ALBPs, the simple assembly line balancing problem (SALBP) is a fundamental problem encountered for optimizing assembly systems (Campana et al., 2022). There are two main SALBP variants in the literature: SALBP-1, which aims to minimize the number of stations for a given cycle time, and SALBP-2, which aims to minimize the cycle time for a fixed number of stations (Campana et al., 2022). Although ALBP has been widely studied, it simplifies industrial reality; therefore, there are many practical limitations. Mixed integer programming (MIP) has been widely favored in the ALBP literature as a traditional mathematical programming method. However, MIP models can be difficult to linearize when solving today's real-life ALBPs. An abundance of current research focuses on variants to reflect realistic features of the ALBP, and various solution procedures, including mathematical programming approaches, heuristic, and metaheuristic algorithms, have been proposed (Zhang et al., 2021).

Under the conventional SALBP, a single-model assembly line is considered. Single-model assembly lines are suitable for producing single products but are less suited to handling high variety requirements. Current market conditions require assembly lines and manufacturing strategies to be highly flexible to meet the variable needs of customers. Due to the high cost of designing an assembly line for any single model, producers try to assemble a set of products on a mixed-model assembly line. This is called the mixed-model assembly line balancing problem (MALBP). In the MALBP, tasks belonging to different product models are assigned to workstations based on their processing times and precedence relationships amongst

* Corresponding author
E-mail amylee@chu.edu.tw (A.H.I. Lee)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)

doi: 10.5267/j.ijiec.2024.10.008

tasks (Li et al., 2023). Due to the current market demand for diversified products and the need for mixed-model assembly lines, much research on the MALBP has been done in recent years. The MALBP can be divided into two classes: MALBP-I, given a cycle time, the objective is to minimize the number of workstations; and MALBP-II, given a number of workstations, the objective is to minimize the cycle time (Simaria & Vilarinho, 2004). The system design of a multi-objective assembly line balancing problem is necessary in assembly manufacturing for a make-to-order environment. For the MALBP, multiple objectives and constraints may need to be considered, and numerous mathematical models have been proposed to solve different types of problems. Some models can obtain exact solutions, especially for small-scale problems. When the problem becomes too complicated, it becomes NP-hard, and it is impossible to obtain optimal solutions using exact methods. Heuristic or metaheuristic algorithms thus have been used to find satisfactory solutions within a reasonable time.

Real-life production processes often consist of various tasks requiring different worker qualifications. There may be a hierarchical workforce structure in which workers' qualification levels are arranged hierarchically. With the downward substitutability in the hierarchy, lower qualified workers can be replaced by higher qualified workers, but not vice versa. Higher qualified workers have higher labor costs but execute tasks in shorter times. In this work, we consider the multi-objective mixed-model assembly line balancing problem with hierarchical worker assignment (MO-MALBP-HW), in which the objectives are minimizing cycle time and minimizing hierarchical workforce cost. The proposed multi-objective programming (MOP) model can solve small-scale problems efficiently. When the problem becomes too complicated, it becomes NP-hard, and the multi-objective genetic algorithm (MOGA) and the non-dominated sorting genetic algorithm II (NSGA-II) can obtain near-optimal solutions for large-scale problems in a short computational time. Case studies of a gear reducer manufacturer and larger cases from previous literature will be presented to examine the practicality of the proposed models for solving the problem. In addition, task assignment plans can be generated as a result. To the authors' knowledge, this is the first work that considers multiple objectives, mixed-model assembly line balancing, and hierarchical worker assignment simultaneously.

The rest of this paper is organized as follows. Section 2 reviews the MALBP and the hierarchical workforce. Section 3 constructs the MO-MALBP-HW using mixed integer programming (MIP), multi-objective genetic algorithm (MOGA), and non-dominated sorting genetic algorithm II (NSGA-II). Some instances are solved using the MIP, the MOGA, and the NSGA-II in Section 4. Conclusions and future research directions are presented in the last section.

2. Mixed-model assembly line balancing problem and hierarchical workforce

Customers' demands for different models and features change constantly in today's competitive business environment. Mass production has shifted to a multi-variety and small-batch production model, and simple assembly lines cannot meet the ever-changing market demand (Liu et al., 2021). Building and maintaining an assembly line for one single model is costly and unrealistic. Thus, a mixed model-based assembly line can avoid constructing several lines while satisfying customer demands and minimizing workers and costs. More and more researchers are studying the mixed-model assembly line balancing problem (MALBP), and many industries are developing mixed-model assembly lines. Razali et al. (2019) studied the recent trend in mixed-model assembly line balancing optimization and reviewed some publications from 2002 to early 2018 that studied various ALBPs. The discussions included problem varieties, optimization algorithms, objective functions, and constraints in the problem. Some works of the MALBP and their approaches were reviewed.

The MALBP has been widely studied recently. Liu et al. (2021) applied the uncertainty theory to model uncertain demand, and a complexity theory was introduced to measure the uncertainty of mixed-model assembly line balancing. Scenario probabilities and triangular fuzzy numbers were used to describe the uncertain demand. Considering the influence of multi-model products on the mixed-model assembly line, the workstation complexity was measured based on information entropy and fuzzy entropy to help balance the system with robust performance. The validity of the model was verified by multiple examples of automotive engine mixed-model assembly lines. Zhang et al. (2021) studied the mixed-model multi-manned assembly line balancing under uncertain demand and aimed to optimize the assembly line configuration through a mixed integer linear programming (MILP) model. In addition, a gene expression programming (GEP) method with multi-attribute representation was developed to generate efficient scheduling rules, and these rules were embedded into solution generation mechanisms to obtain line configuration. Meng et al. (2022) studied the MALBP considering preventive maintenance scenarios. A mixed integer mathematical model was proposed to optimize cycle time and task changes. The co-evolutionary algorithm was applied to simplify large-scale cases with a powerful divide-and-conquer architecture, and four improvements were made. Experimental results showed that the proposed mathematical model could obtain the Pareto solution for small-scale examples. The developed algorithm outperforms other algorithms on three criteria, and the Pareto front obtained by this algorithm was closer to the true Pareto front. Belkharroubi and Yahyaoui (2022) asserted that the assembly line was the main element responsible for assembling products in a manufacturing system, and it need to be well managed to avoid problems that could lead to production failures. MALBP-I occurs when designing a new assembly line, where different models of one product are assembled in a mixed order. The authors proposed a hybrid reactive greedy randomized adaptive search procedure to solve the problem by optimizing the number of workstations for a fixed, known cycle time. Delice et al. (2023) constructed a mixed integer mathematical model to minimize the total costs of the assembly line and supermarkets, and constraint programming (CP) was used to solve the developed model. For large-scale problems, a method based on ant colony optimization (ACO) and simulated annealing (SA) was developed to solve the supermarket location problem and the MALBP simultaneously.

Under hierarchical workforce scheduling (HSW), tasks vary in processing requirements and qualification levels of workers. A higher-skilled worker can replace a lower-skilled worker to perform a task but at a higher cost (Sungur & Yavuz, 2015). In addition, some tasks can only be completed by qualified, highly skilled workers. Sungur and Yavuz (2015) introduced an ALBP where tasks differed in terms of eligibility requirements, and workers' qualification levels were ranked hierarchically. The assembly line balancing with hierarchical worker assignment was solved using an integer linear programming model. Faccio et al. (2016) developed an innovative balancing and sequencing hierarchical method for mixed-model assembly lines using jolly operators with a side-by-side strategy. The goals were to minimize the number of work overloads and the necessary number of jolly operators, and the proposed procedure was validated by numerical analysis and comparison of optimized sequences with standard sequencing rules of thumb. Małachowski and Korytkowski (2016) constructed a competency-based analysis model to analyze the performance of multi-skilled workers taking on repetitive tasks. An analytical tool that better-described workers' ability to perform repetitive tasks was developed by combining hierarchical capabilities, which were modeled as a weighted digraph, and learning curves, which represented individual learning rates. An assembly line model using discrete events was simulated, and how work experience in one job translated into performance in other roles was calculated. A better workforce scheduling could be generated as a result. Korytkowski (2017) developed a competency-based analysis model for analyzing the performance of multi-skilled workers doing repetitive tasks, considering learning and forgetting. The learning curve was used to estimate improvements when repeating the same action. The inverse phenomenon was forgetting, which could occur due to interruptions in the production process. The performance evaluation algorithm was developed for two scenarios: fixed shift duration and fixed production output. Chen et al. (2019) developed a model for the MALBP in the TFT-LCD module process, considering resource constraints, minimizing the number of workstations and workers, optimizing the assignment of tasks, workers and machines, and multi-skilled workers. A mathematical programming model was proposed to obtain the optimal solution, and a heuristic two-stage method based on adaptive GA was developed to solve NP-hard problems. Campana et al. (2022) developed an algorithm for ALBPs with hierarchical worker allocation for firms with multi-skilled workforces. The aim was to assign workers and tasks to workstations on the assembly line to meet cycle time and priority constraints and minimize total costs. A mathematical model was proposed first, and it was improved by preprocessing techniques. A constructive heuristic and a variable neighborhood descent were constructed to solve large instances. Liu et al. (2023) studied a scheduling problem in a hybrid flow shop where all processing stages were composed of unrelated parallel machines, and the objective was to minimize the makespan with the consideration of multi-skilled workers and their fatigue states. An agent-based simulation system was developed to cope with the uncertainties in worker fatigue models. A simulation-based optimization (SBO) framework, which integrated genetic algorithm (GA) and reinforcement learning (RL), was established to solve the problem. A hybrid flow shop examined the feasibility and effectiveness of the SBO approach in a pharmaceutical factory. Alhomaidhi (2024) introduced a mixed-model assembly line balancing approach that incorporated the learning effect alongside specific worker-type requirements for each task. An integer programming model complemented by heuristic techniques, which took into account variables such as cycle time, task dependencies, worker classifications, and the learning curve, were proposed. The goal was to minimize costs related to both labor and workstations while enhancing overall production efficiency.

Most earlier research on mixed-model assembly line problems considered single-objective optimization (Saif et al., 2014). In a real production environment, however, multiple objectives need to be optimized simultaneously. Among the earlier works that tackled multi-objective optimization, the objectives are often combined into a single objective for optimization. For example, Simaria and Vilarinho (2004) studied a MALBP-II for a given number of parallel workstations and some zoning constraints. A mathematical model was first developed with a minimization objective, which was a sum of two terms: cycle time and workload balance within the workstations. A GA-based approach was applied next to solve the problem. Over the past decade, different types of multi-objective algorithms and their extensions in multi-objective optimization have been studied in the literature (Saif et al., 2014). Saif et al. (2014) studied a mixed-model assembly line with simultaneous sequencing and balancing. The multiple objectives were balancing the workload of different models on each station, reducing the deviation of the workload of a station from the average workload of all the stations, and minimizing the total flowtime of models on different stations. A multi-objective artificial bee colony algorithm for simultaneous sequencing and balancing of the mixed-model assembly line with Pareto concepts and local search mechanism was developed. Rabbani et al. (2020) studied a multi-objective particle swarm optimization model, and an augmented multi-objective particle swarm optimization model was developed and compared. The result showed that the latter had a better performance. Chen et al. (2023) studied an integrated assembly line balancing and part-feeding problem. A bi-level mathematical model was proposed first to simultaneously minimize the number of stations and workload smoothness of the assembly line at the upper level and to minimize the number of supermarkets of the part feeding at the lower level. A bi-level multi-objective genetic algorithm was developed next, and it had two strategies for problem-solving: extending fitness evaluation to improve the approximation to the true frontier and adaptive termination condition to accelerate the convergence. Sun et al. (2024) studied a multi-objective hybrid production line balancing problem considering disassembly and assembly. The multiple objectives were to optimize cycle time, total cost, and workload smoothness concurrently under a fixed number of workstations. A mathematical programming model was formulated first, and a Pareto-based hybrid genetic simulated annealing algorithm was developed next. The two-point crossover and hybrid mutation operator were applied to generate potential non-dominated solutions, and a local search method based on a parallel simulated annealing algorithm was used to perform a depth search.

GA has been adopted to solve the multi-objective ALBP. Some recent works are reviewed here. Zhang et al. (2020) studied the MALBP by proposing a multi-objective model to minimize energy consumption and maximize line efficiency. A multi-

objective cellular GA, which integrated a cellular strategy and local search, was developed to solve the problem. Tanhaie et al. (2020) studied a simultaneous balancing and worker assignment problem for mixed-model assembly lines in a make-to-order environment. A multi-objective model, which simultaneously minimized the number of stations and the total cost of the task duplication and worker assignment, was developed, and a control system that minimized the work in process was constructed. An NSGA-II was also developed, and it performed better than four other multi-objective algorithms. Li et al. (2023) proposed a multi-objective GA for the SALBP, called a bespoke genetic algorithm, to consider three objectives: minimizing the number of workstations, minimizing system cycle time, and maximizing workload smoothness. Nourmohammadi et al. (2023) studied a MALBP that incorporates musculoskeletal risk assessment. A MILP model was developed to consider three objectives: minimizing cycle time, minimizing the maximum ergonomic risk of workstations, and minimizing the total ergonomic risk. An enhanced non-dominated sorting genetic algorithm II (E-NSGA-II) was also developed by integrating a local search procedure to generate neighborhood solutions and a multi-criteria decision-making mechanism approach based on TOPSIS to ensure the selection of quality solutions.

3. Research methods

This research constructs a mathematical model for the multi-objective mixed-model assembly line balancing problem with hierarchical worker assignment (MO-MALBP-HW). For large-scale problem instances, both a MOGA approach and an NSGA-II approach are applied.

3.1 Assumptions and notations

The basic assumptions for the MO-MALBP-HW are as follows (Saif et al., 2014; Chen et al., 2019; Zhang et al., 2021; Liu et al., 2021; Belkharroubi and Yahyaoui, 2022; Liu et al., 2023).

Assumptions

1. There is only one planning period.
2. The number of stations in the assembly line is known.
3. All machines and workers are available at the beginning of the period.
4. Several models with similar characteristics are being assembled.
5. The demand for each model is known in advance.
6. The precedence relationships of tasks for each model are known.
7. The task processing times of different models are deterministic and different.
8. All tasks must be assigned.
9. Each task is only assigned to one worker in one station.
10. Each station has only one worker with one hierarchical level.
11. The unit labor cost is different for different worker types.
12. A task of a model with a zero processing time means that the model does not contain this task.

The investigation of the MO-MALBP-HW is based on the following notations, which are modified from those used in some previous works (Özcan & Toklu, 2009; Sungur and Yavuz, 2015; Faccio et al., 2016; Li et al., 2018; Rashid et al., 2020; Lee et al., 2021; Campana et al., 2022; Kang & Lee, 2023).

Notations

Indices:

j, u, v Task (j = 1,2,…, J).
k Station (k = 1,2,…, K).
m Product model *(m = 1,2,...,M).*
h Worker type (h= 1,2,…, H).

Parameters:

$\boldsymbol{J}$ Set of tasks (j = 1,2,…, J).
$\boldsymbol{K}$ Set of stations (k = 1,2,…, K).
$\boldsymbol{M}$ Set of product models (m = 1,2,…, M).
$\boldsymbol{H}$ Set of worker types (h = 1,2,…, H), with the skill level descending as h increases.
w_1 The importance weight for cycle time.
w_2 The importance weight for total cost.
t_{jm}^{h} Processing time of task j of type h for model m.

c_h Unit labor cost of worker type *h*.
Ω Set of precedence relations; $(u,v)\in\Omega$ if and only if task *u* is an immediate predecessor of task *v*.
PZ Set of pairs of tasks in positive zoning; $(u,v)\in PZ$ if and only if tasks *u* and *v* must be processed in the same station.
NZ Set of pairs of tasks in negative zoning; $(u,v)\in NZ$ if and only if tasks *u* and *v* cannot be processed in the same station.

Decision variables:
CT Cycle time.
CH Total cost of hierarchical workforce.
T_{km} Completion time of station *k* for model *m*.
UB Upper bound of the number of stations used.
X_{jk}^{h} A binary variable, equal to 1 if task *j* is processed by worker type *h* in station *k*.
Y_{k}^{h} A binary variable, equal to 1 if station *k* is operated by worker type *h*.
V_k A binary variable, equal to 1 if station *k* is selected for processing.

3.2 Mixed integer programming (MIP) model for the MO-MALBP-HW

The mixed integer programming (MIP) model for solving the multi-objective mixed-model assembly line balancing problem with hierarchical worker assignment (MO-MALBP-HW) is constructed as follows:

$$\min\ f = w_1 \times CT + w_2 \times CH \tag{1}$$

$$\text{subject to } \sum_{h=1}^{H}\sum_{k=1}^{K} X_{jk}^{h} = 1, \qquad j \in \boldsymbol{J} \tag{2}$$

$$\sum_{h=1}^{H}\sum_{k=1}^{K} k \times X_{vk}^{h} - \sum_{h=1}^{H}\sum_{k=1}^{K} k \times X_{uk}^{h} \geq 0, \qquad \forall (u,v) \in \Omega \tag{3}$$

$$T_{km} = \sum_{h=1}^{H}\sum_{j=1}^{J} t_{jm}^{h} \times X_{jk}^{h}, \qquad k \in \boldsymbol{K}, m \in \boldsymbol{M} \tag{4}$$

$$T_{km} \leq CT, \qquad k \in \boldsymbol{K}, m \in \boldsymbol{M} \tag{5}$$

$$\sum_{j=1}^{J} X_{jk}^{h} \leq J \times Y_{k}^{h}, \qquad k \in \boldsymbol{K} \tag{6}$$

$$\sum_{j=1}^{J} X_{jk}^{h} \geq Y_{k}^{h}, \qquad k \in \boldsymbol{K} \tag{7}$$

$$\sum_{h=1}^{H} Y_{k}^{h} \leq H \times V_k, \qquad k \in \boldsymbol{K} \tag{8}$$

$$\sum_{h=1}^{H} Y_{k}^{h} \geq V_k, \qquad k \in \boldsymbol{K} \tag{9}$$

$$V_k \geq V_{k+1} \qquad k \in \boldsymbol{K} \tag{10}$$

$$K \geq \sum_{k=1}^{K} V_k \tag{11}$$

$$CH = \sum_{h=1}^{H}\sum_{k=1}^{K} c_h \times Y_{k}^{h} \tag{12}$$

$$X_{uk}^{h} - X_{vk}^{h} = 0 \qquad \forall (u,v) \in PZ \tag{13}$$

$$X_{uk}^{h} + X_{vk}^{h} \leq 1 \qquad \forall (u,v) \in NZ \tag{14}$$

$$X_{jk}^{h} \in \{0,1\} \qquad j \in \boldsymbol{J}, k \in \boldsymbol{K}, h \in \boldsymbol{H} \tag{15}$$

$$Y_{k}^{h} \in \{0,1\} \qquad k \in \boldsymbol{K}, h \in \boldsymbol{H} \tag{16}$$

$$V_k \in \{0,1\} \qquad k \in \boldsymbol{K} \tag{17}$$

where objective function (1) is a multi-objective function, which minimizes the cycle time with weight w_1 and minimizes the total cost of the hierarchical workforce with weight w_2. Constraints (2) ensure that a task can only be assigned and processed by one single station with one worker of type h. Constraints (3) ensure the sequencing of tasks; a task needs to be completed before its next task can proceed. Constraints (4) calculate the completion time of station k for model m, T_{km}, by summing up the processing times of all tasks for model m in processing stations. Constraints (5) let the completion time of station k for model m be less than or equal to the cycle time. Constraints (6) and (7) ensure that station k is operated by a worker with type h if task j is being processed by a worker with type h in that station. Constraints (8) and (9) ensure that a worker with type h is operating in station k if the worker of type h is processing in that station and only one worker is working in that station. Constraints (10) allow station k+1 to be used only if preceding station k is used. Constraints (11) ensure the total number of stations used, K, is larger than the sum of all V_k's. Constraints (12) calculate the total cost of the hierarchical workforce. Constraints (13) ensure that tasks u and v must be processed in the same station. Constraints (14) ensure that tasks u and v cannot be processed in the same station. Constraints (15) show that X_{jk}^h is a binary variable, equal to 1 if task j for type h is processed in station k. Constraints (16) show that Y_k^h is a binary variable, equal to 1 if station k for type h is selected for processing. Constraints (17) show that V_k is a binary variable, equal to 1 if station k is selected for processing. The MOGA will solve the above MIP model. Under the NSGA-II, the objective weights are not assigned. The mathematical model is constructed as follows:

$$\min \ f_1 = CT \tag{18}$$

$$\min \ f_2 = CH \tag{19}$$

$$\text{subject to} \quad \text{constraints (2)–(17)} \tag{20}$$

where objective function (18) is to minimize the cycle time and objective function (19) is to minimize the total cost of the hierarchical workforce. Constraint (20) contains all the constraints, i.e. constraints (2) to (17), in the previous MIP model.

3.3 MOGA for the MO-MALBP-HW

Genetic algorithm (GA) is a heuristic search algorithm commonly used to solve optimization problems. A two-point crossover and a uniform mutation operator are applied to prevent a solution from being confined to a local optimum and advancing toward the global optimum. The flowchart of the MOGA process is shown in Fig. 1. The steps of the process are as follows (Simaria and Vilarinho, 2004; Zhang et al., 2019; Lee et al., 2021; Kang & Lee, 2023; Liao et al., 2023):

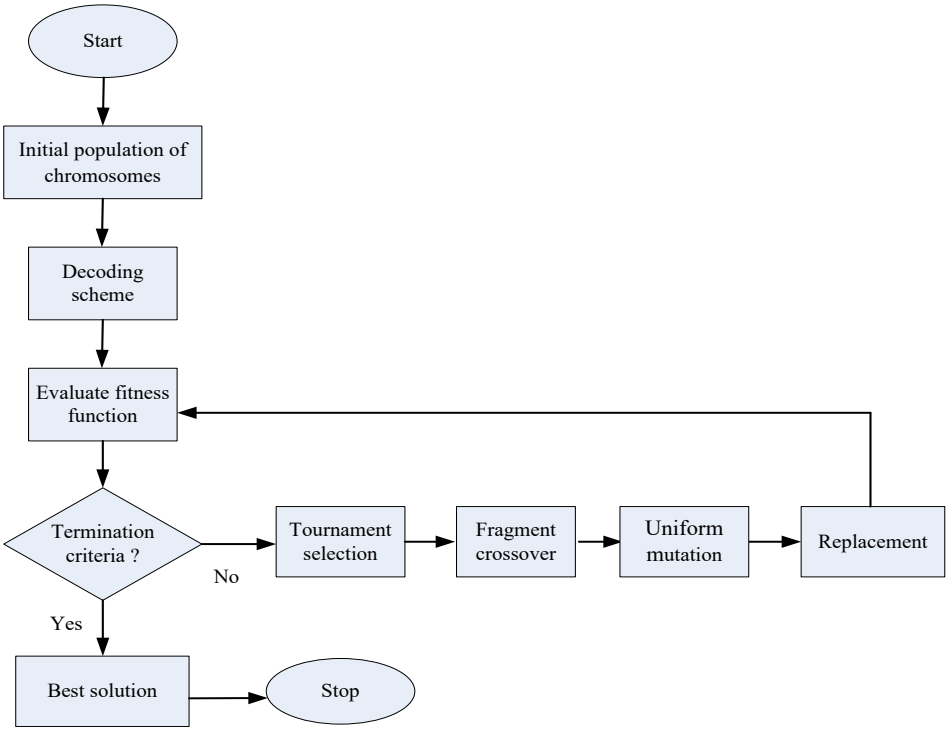


Fig. 1. Graphical representation of MOGA process

Step 1. Initial population of chromosomes

Before performing the GA operations, a certain number of chromosomes must be randomly obtained. A collection of chromosomes is called a population, and the population size usually depends on the complexity of the problem. The more complex the problem, the larger the required population and the stronger its ability. As a result, the solution is less likely to fall into the local optimal solution. It is assumed that a job can only be assigned to one station and one worker type. With a population size of N, J tasks and K stations, the ith chromosome has two fragments, $Chrom^i\_s$ and $Chrom^i\_t$, represented as a string of J and K integer digits, respectively. $Chrom^i\_s$ has J genes, with each jth gene representing a random ordering of station k (1 to K) for processing task j. $Chrom^i\_t$ has K genes, with each kth gene representing a random ordering of worker type h (1 to H) for station k. Fig. 2 shows an example of a chromosome population of size N with J tasks and K stations.

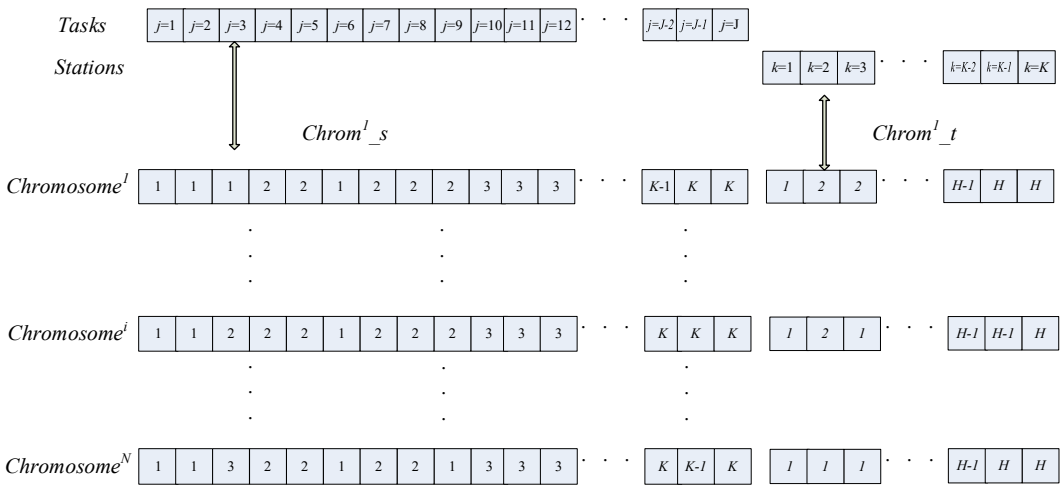


Fig. 2. Population of chromosomes and chromosome of coding scheme

Step 2. Decoding scheme

Each chromosome can be decoded as one single solution. Use $Chrom^N$ in Fig. 2 as an example, and the decoding scheme is as shown in Fig. 3. $Chrom^N\_s$ shows the station for processing task j, and $Chrom^N\_t$ shows the worker type for station k. In $Chrom^N\_s$, tasks 1, 2 and 6 have digit 1, tasks 3, 4, 5, 7, 8, 9 have digit 2, and tasks 10, 11, 12 have digit 3. This indicates that tasks 1, 2 and 6 are assigned to station 1, tasks 3, 4, 5, 7, 8, 9 are assigned to station 2, and tasks 10, 11, 12 are assigned to station 3. The same procedure applies to $Chrom^N\_t$. Stations 1 and 3 have digit 1, and station 2 has digit 2. This indicates that worker type I is assigned to stations 1 and 3, and worker type II is assigned to station 2. The results show that stations 1 and 3 will be operated by worker type I, and station 2 will be operated by worker type II.

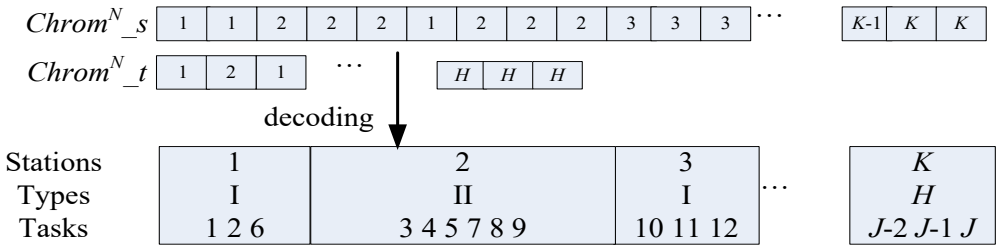


Fig. 3. Chromosome of decoding scheme and for station and work type assignments

Step 3. Fitness function evaluation

Define the fitness function for each chromosome as Max $1/f$, where f is the multi-objective function to be minimized.

Step 4. Tournament selection

Individuals (chromosomes) are selected from the current population to become parents of the next generation. Some chromosomes are randomly selected based on their fitness; that is, the chromosomes with the best objective values are selected for reproduction. The tournament selection strategy is applied as follows (Goldberg et al., 1989; Zhang et al., 2019):

Step 4.1 A certain amount of chromosomes are randomly selected from the population. In this research, 10% of population size N is selected.

Step 4.2 Choose the chromosome with the best objective value from the selected chromosomes as the best chromosome and add it to the mating pool. All the selected chromosomes in Step 4.1 are returned to the population for the next selection.

Step 4.3 Repeat this process until the number of chromosomes in the mating pool reaches population size N, and the population is updated through this process.

Step 5. Fragment crossover

Create new individuals (offspring) by combining the genes of selected parents. Various crossover techniques exist, such as single-point crossover, two-point crossover, or uniform crossover. Here, we use the two-point crossover technique, and the steps are as follows:

Step 5.1 Two parent individuals are selected from the population.

Step 5.2 Each of the two parents is divided by two randomly cut points, one in each fragment.

Step 5.3 For offspring 1, the first part in the first fragment is taken from parent 2, and the second part in the first fragment is taken from parent 1. The first part of the second fragment is taken from parent 1, and the second part of the second fragment is taken from parent 2.

Step 5.4 For offspring 2, the first part in the first fragment is taken from parent 1, and the second part in the first fragment is taken from parent 2. The first part of the second fragment is taken from parent 2, and the second part of the second fragment is taken from parent 1.

Step 5.5 Repeat the process until N chromosomes are collected.

An example of the fragment crossover is demonstrated in Fig. 4.

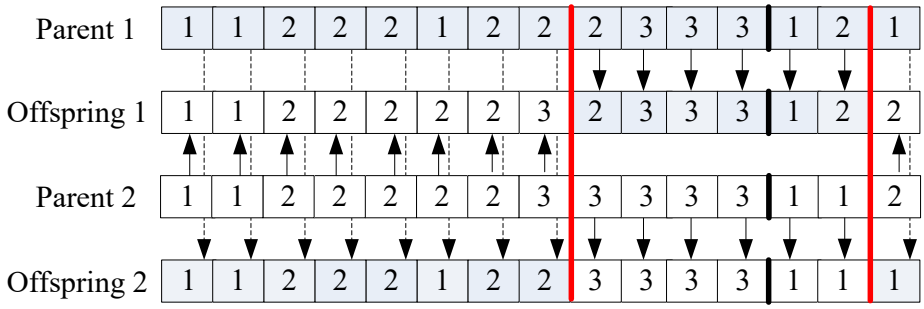


Fig. 4. A fragment crossover example with a two-point crossover

Step 6. Uniform mutation

Mutation makes random changes to some genes in offspring. Uniform mutations help introduce genetic diversity into a population and prevent the algorithm from falling into local optima. Fig. 5 shows an example of the mutation of offspring 1, as some genes in the first and second fragments are mutated.

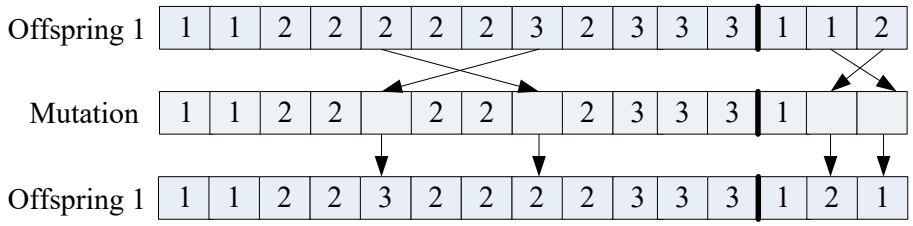


Fig. 5. A mutation example

Step 7. Replacement

A new population is obtained after completing the fragment crossover and mutation operations in a generation. The next generation is created by replacing the old population with offspring. The population size remains the same.

Step 8. Termination

The replacement operation and new population generation are repeated until a fixed number of generations or until a termination condition is met. In this study, the algorithm is terminated when the generation (Gen) reaches G_{max}.

Step 9. Output

The best individual (chromosome) found in the last generation is considered the best solution to the problem.

3.4 NSGA-II for the MO-MALBP-HW

In NSGA-II, solutions are evaluated based on multiple objective functions, and their fitness values are determined through non-dominated sorting and crowding distance calculation processes. The flowchart of the NSGA-II process is shown in Fig. 6. Some of the steps of the NSGA-II are similar to those of the MOGA introduced in section 3.3. The steps of the process are briefly presented as follows (Deb et al., 2002; Chutima & Khotsaenlee, 2022; Nourmohammadi et al., 2023; Rahman et al., 2023; Samouei & Sobhishoja, 2023; Soysal-Kurt et al., 2024):

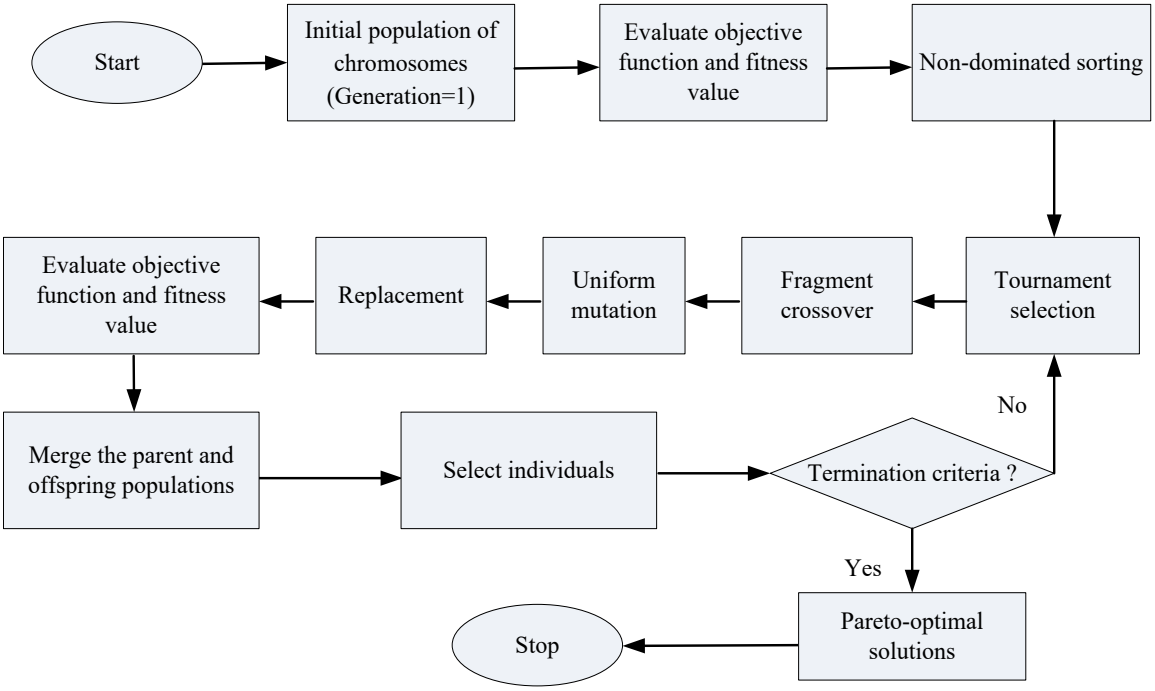


Fig. 6. Graphical representation of NSGA-II process

Step 1. Initial population of chromosomes. This step is similar to that in the MOGA.

Step 2. Evaluate objective function and fitness value. The objective functions for each chromosome are Equation (18), minimizing the cycle time, and Equation (19), minimizing the total cost of the hierarchical workforce. The fitness value is determined through non-dominated sorting and crowding distance calculation.

Step 3. Non-dominated sorting. This step includes initializing the number of solutions that dominate this solution and a list of solutions that this solution dominates, computing domination count and dominated solutions, identifying the first front, and iteratively identifying subsequent fronts.

Step 4. Tournament selection. This step is similar to that in the MOGA.

Step 5. Fragment crossover. This step is similar to that in the MOGA.

Step 6. Uniform mutation. This step is similar to that in the MOGA.

Step 7. Replacement. This step is similar to that in the MOGA.

Step 8. Evaluate objective function and fitness value. After Steps 3 to 7, the objective function and fitness value are re-evaluated.

Step 9. Merge the parent and offspring populations. This step includes initializing populations, combining populations, non-dominated sorting, and selecting solutions for the next generation.

Step 10. Select individuals. Individuals are selected based on ranking and crowding distance.

Step 11. Termination. This step is similar to that in the MOGA.

Step 12. Output. The solutions on the Pareto frontier are obtained. These are the Pareto-optimal solutions.

4. Case studies

Five case studies are carried out to examine the proposed MIP model, the MOGA approach, and the NSGA-II approach. The MIP model can obtain the optimal solutions for small and medium-sized problem instances (Cases 1, 2, and 3). The MOGA and the NSGA-II approaches can generate optimal solutions for small problems, i.e., Cases 1 and 2, and near-optimal solutions for medium-sized problems, i.e., Case 3. For large-sized problems (Case 4 and Case 5), the problem becomes NP-hard, and the MIP model can no longer solve the problem. However, the MOGA and NSGA-II approaches can still obtain near-optimal solutions. In addition, the data for Case 2 and Case 3 is from a gear reducer manufacturer in Taiwan. The data for Case 4 is extracted and revised from Tonge (1961) and Li et al. (2018). The data for Case 5 is extracted and revised from Bartholdi (1993), Özcan and Toklu (2009) and Campana et al. (2022).

Commercial programs are applied to solve the models. LINGO (2018) is used to solve the MIP model, and MATLAB (2019) is used to solve the MOGA and NSGA-II models. The models are executed on a PC with an Intel® Core ™ i7, 3.60GHz processor, and 4GB RAM. The MIP, the MOGA, and the NSGA-II calculate the objective values for problem scenarios. Gaps between the results of the MIP and the MOGA are calculated by Equation (21).

$$\%\ \text{Gap} = \frac{\text{MOGA objective value} - \text{MIP objective value}}{\text{MIP objective value}} = \frac{f_{\text{MOGA}} - f_{\text{MIP}}}{f_{\text{MIP}}} \tag{21}$$

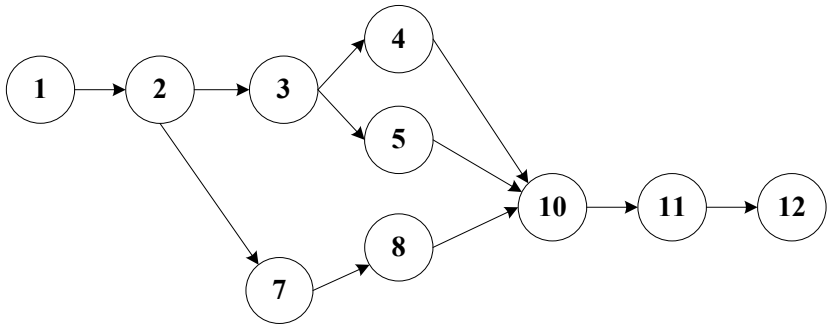
where f_{MIP} is the objective value obtained by the MIP, and f_{MOGA} is the objective value obtained by the MOGA.

4.1 Case 1

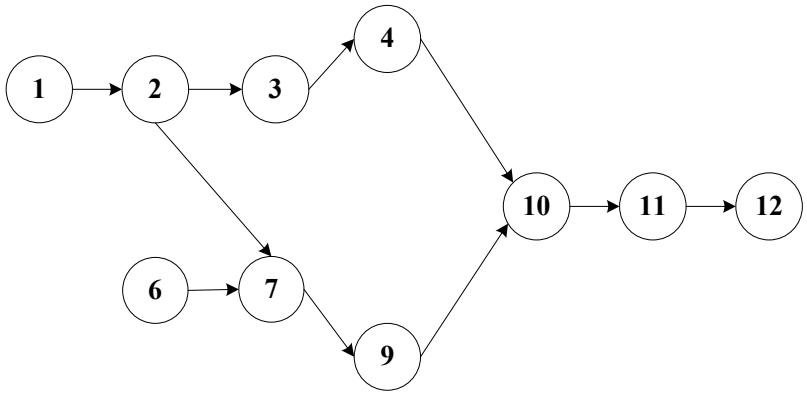
In Case 1, the MO-MALBP-HW is applied to solve a problem with two product models, comprising 12 tasks. The assignment of assembly tasks into stations is constrained by precedence relationships, which are shown in a diagram called a precedence diagram. In the MALBP, multiple similar product models are assembled simultaneously. Each model has its own specific precedence relationships, and processing times may vary among models. Due to the similarity among these models, the tasks of these models can be combined into a precedence diagram, called a combined precedence diagram. The precedence diagrams of the two models and the combined precedence diagram are shown in Fig. 7. The task information of the 12 tasks is shown in Table 1, which lists the immediate predecessor(s) of each task and the processing time of each task for each of the two models by type I worker and type II worker. Note that "0" indicates that the model does not need the specific task, and "+∞" indicates that a worker with the specific type is not qualified to perform the task. We assume that there are two types of workers: type I and type II. Type I workers are highly skilled and thus can perform tasks quickly, and the unit cost for a type I worker is \$350. Type II workers have fewer skills than highly skilled workers; thus, they need more time to perform tasks. The unit cost for a type II worker is \$300. Note that type II workers cannot process some of the tasks.

Table 1
Task information for Case 1

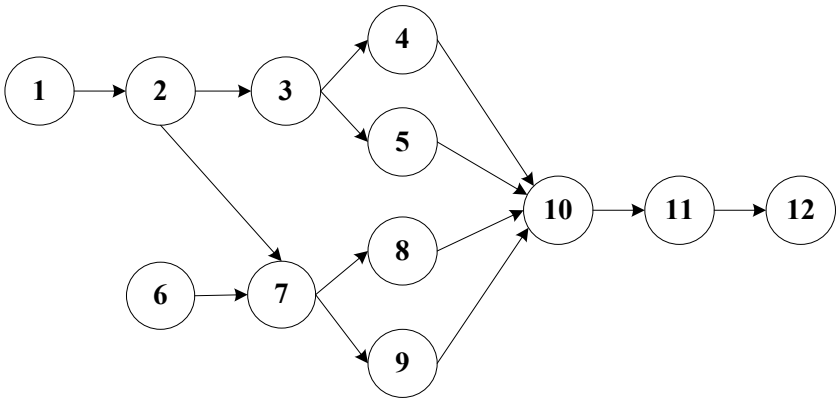
Task	Predecessors	Processing time by type I worker (seconds)		Processing time by type II worker (seconds)	
		Model A	Model B	Model A	Model B
1		270	240	450	400
2	1	90	90	150	150
3	2	60	60	100	100
4	3	150	0	250	0
5	3	90	120	150	200
6		0	600	0	100
7	2, 6	60	120	100	200
8	7	60	0	100	0
9	7	0	120	0	200
10	4, 5, 8, 9	60	90	+∞	+∞
11	10	150	120	+∞	+∞
12	11	90	150	+∞	+∞



(a) model I



(b) model II



(c) combined

Fig. 7. Precedence diagrams of model I, model II and combined

4.1.1 Case 1 with Objective Weights

The MIP and the MOGA can be applied when distinct objective weights are assigned. Let the weight of objective 1 (w_1), the minimization of cycle time (CT), be 1, and the weight of objective 2 (w_2), the minimization of the total cost of the hierarchical workforce (CH), be 10^{-6}. That is, cycle time is much more important than the total cost of the hierarchical workforce. Three stations are available. Table 2 shows the assignment results. Each station has one highly skilled worker (worker type I). The total labor cost is 3*$350=$1,050, and the optimal cycle time is 420 seconds.

Table 2
Optimal assignment for Case 1 with N=3, w_1=1 and w_2=10^{-6}

			Model A		Model B	
Station	Tasks	Worker types	Tasks	Station time	Tasks	Station time
1	1, 2, 6	I	1, 2	360	1, 2, 6	390
2	3, 4, 5, 7, 8, 9	I	3, 4, 5, 7, 8	420	3, 5, 7, 9	420
3	10, 11, 12	I	10, 11, 12	300	10, 11, 12	360

On the other hand, assume that the total cost of a hierarchical workforce is much more important than cycle time. Let w_1 be 10^{-6} and w_2 be 1. Table 3 shows the assignment results. Stations 1 and 2 are operated by skilled workers (worker type II), and station 3 is operated by a highly skilled worker (worker type I). The total labor cost is 2*$300+1*$350=$950, and the optimal cycle time is 600 seconds.

Table 3
Optimal assignment for Case 1 with N=3, w_1=10^{-6} and w_2=1

			Model A		Model B	
Station	Tasks	Worker types	Tasks	Station time	Tasks	Station time
1	1, 2	II	1, 2	600	1, 2	550
2	3, 4, 6, 7, 9	II	3, 4, 7	450	3, 6, 7, 9	600
3	5, 8, 10, 11, 12	I	5, 8, 10, 11, 12	450	5, 10, 11, 12	480

Under Case 1, there are two product models and two worker types (worker types I and II). Let either $w_1 = 1$ and $w_2 = 10^{-6}$ or $w_1 = 10^{-6}$ and $w_2 = 1$. The upper bound of the stations can be determined as follows:

$$UB = max\left\{\left\lceil \frac{\sum_{j=1}^{J} t_{j1}^{h}}{max\left(t_{j1}^{h}\right)} \right\rceil, \left\lceil \frac{\sum_{j=1}^{J} t_{j2}^{h}}{max\left(t_{j2}^{h}\right)} \right\rceil, \ldots, \left\lceil \frac{\sum_{j=1}^{J} t_{jM}^{h}}{max\left(t_{jM}^{h}\right)} \right\rceil\right\} \quad (22)$$

Based on the information in Case 1, the calculation shows that UB is 5. Therefore, let the number of stations (K) be 5, 4, 3 or 2. Population size (N) is set to 10. The crossover rate (P_c) is usually set to 0.6 to 1.0 (Lin *et al.*, 2003). In Case 1, it is set to 0.8. The mutation rate (P_m) is usually set to a value less than 0.1 (Lin *et al.*, 2003). In Case 1, it is set to 0.05. The MOGA is terminated when the 100th generation (G_{max}) is attained. The process of each scenario is repeated three times, and the results with the best objective value are collected.

The optimal solutions under each scenario using the MIP and the MOGA are shown in Table 4. For example, under scenario 1, there are five stations (K =5), w_1 = 1 and w_2 = 10^{-6}; the optimal solutions under the MIP and the MOGA are the same, with cycle time (CT) of 270 seconds, the total cost of the hierarchical workforce (CH) of $1,750, and the objective value of 270.0018. The computational time for the MIP is 0.51 seconds, and for the MOGA is 1.24 seconds. Fig. 8 shows that the convergence of MOGA with K=3 is reached at the 12th generation in scenario 3. The optimal assignments for three stations in Tables 2 and 3 are listed as scenarios 3 and 7, respectively, in Table 4. As can be seen in Table 4, the optimal solutions under the MIP and under the MOGA are the same in each scenario, with the %Gap equal to zero. Based on the proposed models, the management can make proper assignment decisions. For instance, if the production line has K stations and the management determines the weights of the two objectives to be w_1 and w_2, the tasks and worker types for each station can be determined by the proposed models, as are the total labor cost and the optimal cycle time.

Table 4
Results for Case 1 using MIP and MOGA

Scenario	K	w_1	w_2	CT	CH	f	MIP CPU(s)	MOGA CPU(s)	MOGA %Gap
1	5	1	10^{-6}	270	1750	270.0018	0.51	1.24	0
2	4	1	10^{-6}	330	1400	330.0014	0.63	1.25	0
3	3	1	10^{-6}	420	1050	420.0011	0.54	1.21	0
4	2	1	10^{-6}	600	700	600.0007	0.33	1.29	0
5	5	10^{-6}	1	450	1550	1550.0005	0.44	1.27	0
6	4	10^{-6}	1	500	1250	1250.0005	0.34	1.25	0
7	3	10^{-6}	1	600	950	950.0006	0.43	1.14	0
8	2	10^{-6}	1	850	650	650.0009	0.53	1.26	0

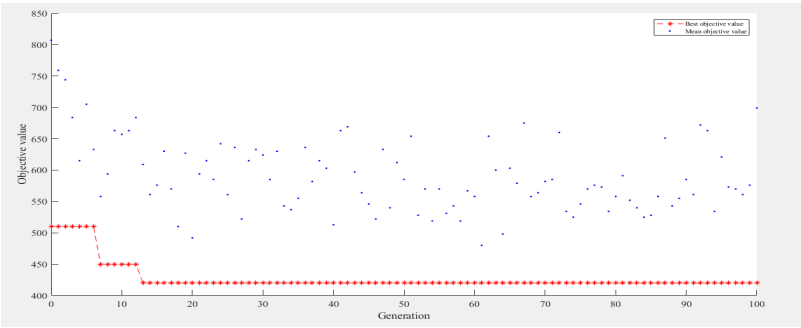


Fig. 8. The convergence of MOGA in scenario 3 for Case 1

4.1.2 *Case 1 without Objective Weights*

When the objective weights are not assigned, the NSGA-II can be applied. In this case, the population size (N) is set to 5, the maximum generation (G_{max}) is set to 100, the crossover rate (P_c) is set to 0.9, and the mutation rate (P_m) is set to 0.05. The results for Case 1 using NSGA-II are shown in Table 5. Scenarios 9, 10, 11, and 12 consider 5, 4, 3, and 2 stations (K), respectively. Under Scenario 9 with five stations, five Pareto optimal solutions are found, such as cycle time (CT) of 270 seconds and total cost of the hierarchical workforce (CH) of \$1,750 as one of the solutions. The computational time is 3.59 seconds. Fig. 9 shows the convergence of the NSGA-II in scenario 11 with K=3 under three dimensions, i.e. cycle time, total cost of hierarchical workforce, and number of stations. Let K=3, the convergence of the NSGA-II in scenario 11 under two dimensions can be seen in Fig. 10. There are three Pareto optimal assignments (cycle time, total cost of hierarchical workforce): (420, 1050), (480, 1000) and (600, 950). Note that in Table 4, both scenarios 3 (w_1 = 1 and w_2 = 10^{-6}) and 7 (w_1 = 10^{-6} and w_2 = 1) are for K=3, and the optimal solution is (420, 1050) and (600, 950), respectively. Since no weights are given to the two objectives under the NSGA-II, there are three solutions on the Pareto frontier. The additional solution under the NSGA-II is (480, 1000).

Table 5
Results for Case 1 using NSGA-II

Scenario	K	(CT, CH)	CPU(s)
9	5	(270, 1750) (330, 1700) (360, 1650) (390, 1600) (450, 1550)	3.59
10	4	(330, 1400) (390,1350) (480,1300) (500, 1250)	3.01
11	3	(420, 1050) (480, 1000) (600, 950)	2.85
12	2	(600, 700) (850, 650)	2.89

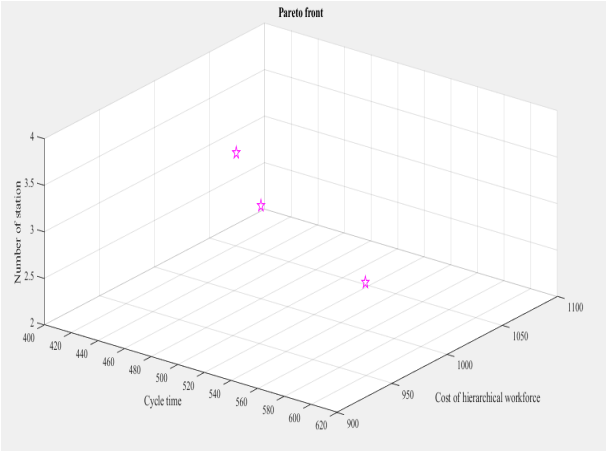


Fig. 9. The convergence of NSGA-II in scenario 11 for Case 1 under 3D

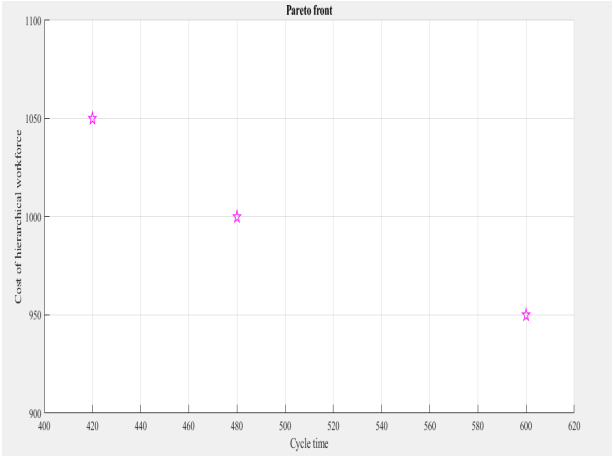
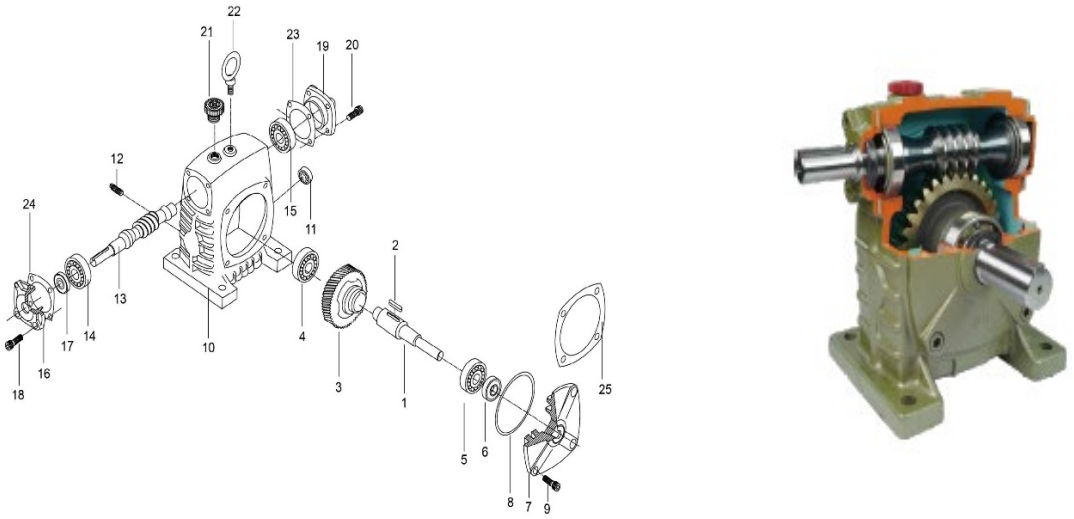


Fig. 10. The convergence of NSGA-II in scenario 11 for Case 1 under 2D

4.2 *Case 2*

In Case 2, the data from an anonymous gear reducer manufacturer in Taichung, Taiwan is used. The company, established in 1969, was specialized in design, R&D, and manufacturing of a wide range of high-tech gear motors, helical gear reducers, worm gear reducers, and planetary gear reducers. In this case, we assume that three product models are produced. The assembly diagram of a gear-worm reducer and final product are shown in Fig. 11. There are a total of 25 tasks. Two types of workers are available: type I (highly skilled) and type II (skilled). The unit cost for a type I worker is \$350, and the unit cost for a type II worker is \$300. The task information of the 25 tasks, including the immediate predecessor(s) of each task and the processing time of each task for each of the three models by type I and type II workers, is shown in Table 6. For the MOGA, population size (N) is set to 50, and G_{max} is set to 300. The crossover rate (P_c) is set to 0.85, and the mutation rate (P_m) is to 0.01. Since the problem in Case 2 is more complex than the one in Case 1, population size (N) and stop generation (G_{max}) are set to be larger numbers. The process of each scenario is repeated three times, and the results with the best objective value are collected. The upper bound of stations (UB) is calculated to be 10; thus, the number of stations (K) ranges from 2 to 10. The solutions under each scenario using the MIP and the proposed MOGA are shown in Table 7. For example, in scenario 7, there are four stations (K =4). With weights w_1 = 1 and w_2 = 10^{-6}, the optimal solutions under the MIP and the MOGA are the same, with cycle time (CT) of 340 seconds, the total cost of the hierarchical workforce (CH) of \$1,400, and the objective value of 340.0014. The computational time for the MIP is 271.98 seconds. The computational time for the MOGA is 33.69 seconds. In another example, in scenario 16, there are also four stations (K =4). The optimal solution (CT, CH) under the MIP is (465, 1250), but the solution under the MOGA is (495, 1250). However, because w_1 = 10^{-6}, CT contributes negligibly to the objective value. Thus, f under the MIP and the MOGA are the same (1250.0005), and %Gap is

zero. The MIP can generate optimal solutions for all the scenarios. While the MOGA can generate optimal solutions for half the scenarios, near-optimal solutions are obtained for the other half of the scenarios, with %Gap of zero.



a. Assembly diagram b. Final product

Fig. 11. Assembly diagram of gear-worm reducer for Case 2

Table 6
Task information for Case 2

Task	Operation	Predecessors	Processing time by type I worker (seconds)		Processing time by type II worker (seconds)	
			Model A	Model B	Model A	Model B
1	Output shaft	-	60	60	90	90
2	Key assemble	1,3	120	90	+∞	+∞
3	Worm wheel	-	60	60	+∞	+∞
4	Ball bearing examine	2	30	30	45	45
5	Ball bearing assemble	4	50	50	75	75
6	Oil seal 1	5	30	30	45	45
7	Output shaft cover	5	60	60	90	90
8	O-Ring assemble	6,7	90	50	135	75
9	Hex screw 1	25	60	60	90	90
10	Outer shell	9	90	80	135	120
11	Oil gauge	9	60	60	90	90
12	Oil drain plug	10,11	120	120	180	180
13	Worm shaft	12	60	60	90	90
14	Taper roller bearing examine	13	30	0	45	0
15	Taper roller bearing assemble	14	50	0	75	0
16	Input shaft cover	14	60	60	90	90
17	Oil seal 2	14	70	60	105	90
18	Hex screw 2	24	50	50	75	75
19	Input shaft cover	15	70	60	105	90
20	Hex screw 3	19,23	50	50	75	75
21	Oil plug	20	15	20	23	30
22	Final hoisting ring	21	20	0	30	0
23	Asbestos free gasket 1	15	10	10	15	15
24	Asbestos free gasket 2	16,17	10	10	15	15
25	Asbestos free gasket 3	8	10	10	15	15

For the NSGA-II, the population size (N) is set to 30, the maximum generation (G_{max}) is set to 300, the crossover rate (P_c) is set to 0.8, and the mutation rate (P_m) is set to 0.01. The result for K=4 is shown in Table 8. The computational time is 45.59 seconds. Fig. 12 shows the convergence of the NSGA-II with K=4 under three dimensions, i.e. cycle time, total cost of hierarchical workforce, and number of stations. The convergence of the NSGA-II under two dimensions can be seen in Fig. 13. There are four Pareto optimal assignments (CT, CH): (340, 1400), (370, 1350), (405, 1300) and (465, 1250). Note that for the MIP, in Table 7, both scenarios 7 ($w_1 = 1$ and $w_2 = 10^{-6}$) and 16 ($w_1 = 10^{-6}$ and $w_2 = 1$) are for K=4, and the optimal solution is (340, 1400) and (465, 1250), respectively. Since no weights are given to the two objectives under the NSGA-II, four solutions are on the Pareto frontier. The two optimal solutions under the MIP are the same under the NSGA-II. The additional solutions under the NSGA-II are (370, 1,350) and (405, 1,300).

Table 7
Results for Case 2 using MIP and MOGA

Scenario	K	w_1	w_2	MIP				MOGA				%Gap
				CT	CH	f	CPU(s)	CT	CH	f	CPU(s)	
1	10	1	10^{-6}	160	3500	160.0035	104.36	160	3500	160.0035	29.02	0
2	9	1	10^{-6}	170	3150	170.0032	631.23	170	3150	170.0032	31.21	0
3	8	1	10^{-6}	200	2800	200.0028	522.15	200	2800	200.0028	32.25	0
4	7	1	10^{-6}	210	2450	210.0025	501.69	210	2450	210.0025	33.63	0
5	6	1	10^{-6}	250	2100	250.0021	591.95	250	2100	250.0021	37.01	0
6	5	1	10^{-6}	300	1750	300.0018	388.56	300	1750	300.0018	32.06	0
7	4	1	10^{-6}	340	1400	340.0014	271.98	340	1400	340.0014	33.69	0
8	3	1	10^{-6}	490	1050	490.0011	165.45	490	1050	490.0011	34.52	0
9	2	1	10^{-6}	675	700	675.0007	121.98	675	700	675.0007	33.12	0
10	10	10^{-6}	1	233	3050	3050.0002	184.65	245	3100	3100.0002	29.98	0
11	9	10^{-6}	1	240	2750	2750.0002	697.36	260	2750	2750.0003	31.26	0
12	8	10^{-6}	1	255	2450	2450.0003	636.35	275	2450	2450.0003	36.39	0
13	7	10^{-6}	1	308	2150	2150.0003	581.45	310	2150	2150.0003	38.59	0
14	6	10^{-6}	1	330	1850	1850.0003	480.56	350	1900	1900.0004	33.45	0
15	5	10^{-6}	1	390	1550	1550.0004	369.69	410	1550	1550.0004	39.68	0
16	4	10^{-6}	1	465	1250	1250.0005	273.45	495	1250	1250.0005	35.78	0
17	3	10^{-6}	1	608	950	950.0006	180.78	630	950	950.0006	34.89	0
18	2	10^{-6}	1	923	650	650.0009	128.98	990	650	650.0010	37.23	0

Table 8
Results for Case 2 with K=4 using NSGA-II

K	(CT, CH)	CPU(s)
4	(340, 1400) (370, 1350) (405, 1300) (465, 1250)	45.59

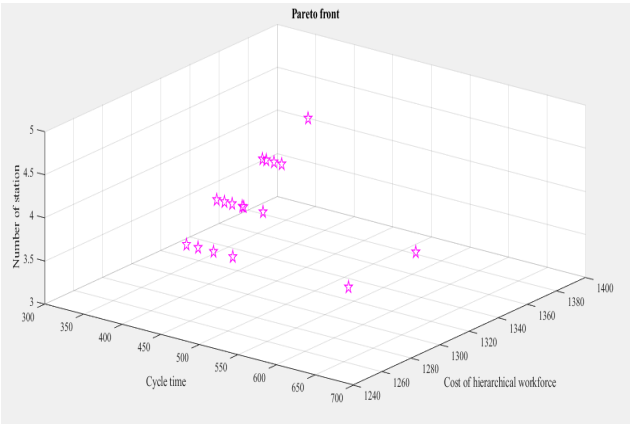


Fig. 12. The convergence of NSGA-II with K=4 for Case 2 under 3D

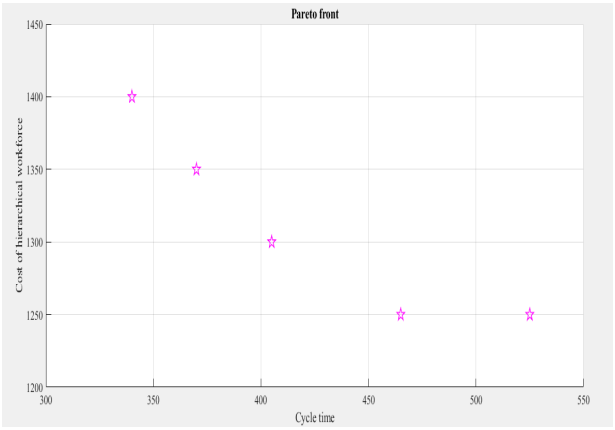


Fig. 13. The convergence of NSGA-II with K=4 for Case 2 under 2D

4.3 Case 3

In Case 3, the data from an anonymous gear reducer manufacturer in Taichung, Taiwan, is also used. The task information of 42 tasks, including the immediate predecessor(s) of each task and the processing time of each task for each model by each type of worker, is shown in Table A1 in the Appendix. Three product models are produced. Two types of workers are available: type I (highly skilled) and type II (skilled). For the MOGA, population size (N) is set to 50, and G_{max} is set to 300. The crossover rate (P_c) is set to 0.8, and the mutation rate (P_m) is set to 0.1. The process of each scenario is repeated three times, and the results with the best objective value are collected.

The calculation shows that the upper bound of stations (UB) is 12. Therefore, let the number of stations (K) range from 2 to 12. The solutions under each scenario using the MIP and the proposed MOGA are shown in Table 9. For instance, scenario 8 has five stations (K =5). With weights w_1 = 1 and w_2 = 10^{-6}, the MIP has the optimal solution with a cycle time (CT) of 720 seconds and a total cost of the hierarchical workforce (CH) of $1,750, and the objective value is 720.0018. The computational time for the MIP is 488.98 seconds. The MOGA has a solution with a cycle time (CT) of 730 seconds and a total cost of the hierarchical workforce (CH) of $1,750, and the objective value is 730.0018. The computational time for the MOGA is 41.92 seconds. The %Gap in scenario 8 is 1.39.%. In another example, in scenario 19, there are also five stations (K =5). The optimal solution (CT, CH) under the MIP is (1050, 1550), but the solution under the MOGA is (1070, 1550). However, because w_1 = 10^{-6}, CT contributes negligibly to the objective value. Thus, f under the MIP and the MOGA are the same (1550.0011), and %Gap is zero. The results in Table 9 show that while the MIP generates optimal solutions, the MOGA can

obtain solutions that are near the optimal ones. Among the 22 scenarios studied, %Gap is zero under 13 of them, and the largest %Gap is 5.88%. Note that the computational time by the MIP can be relatively long under some scenarios.

Table 9
Results for Case 3 using MIP and MOGA

Scenario	K	w_1	w_2	MIP				MOGA				%Gap
				CT	CH	f	CPU(s)	CT	CH	f	CPU(s)	
1	12	1	10^{-6}	330	4200	330.0042	109.52	330	4200	330.0042	41.12	0
2	11	1	10^{-6}	330	3850	330.0039	158.36	330	3900	330.0039	39.61	0
3	10	1	10^{-6}	360	3500	360.0035	184.98	360	3500	360.0035	38.63	0
4	9	1	10^{-6}	420	3150	420.0032	1631.25	425	3200	425.0032	40.69	1.19
5	8	1	10^{-6}	450	2800	450.0028	922.25	455	2800	455.0028	41.83	1.11
6	7	1	10^{-6}	510	2450	510.0025	901.54	540	2450	540.0025	42.91	5.88
7	6	1	10^{-6}	600	2100	600.0021	591.25	630	2100	630.0021	41.77	5
8	5	1	10^{-6}	720	1750	720.0018	488.98	730	1750	730.0018	41.92	1.39
9	4	1	10^{-6}	900	1400	900.0014	371.12	930	1450	930.0015	42.81	3.33
10	3	1	10^{-6}	1200	1050	1200.0011	265.91	1200	1050	1200.0011	44.02	0
11	2	1	10^{-6}	1770	700	1770.0007	151.05	1770	700	1770.0007	42.83	0
12	12	10^{-6}	1	550	3650	3650.0006	189.08	550	3650	3650.0006	40.89	0
13	11	10^{-6}	1	550	3350	3350.0006	17924	560	3350	3350.0006	39.71	0
14	10	10^{-6}	1	550	3050	3050.0006	184.09	580	3100	3100.0006	38.66	1.64
15	9	10^{-6}	1	600	2750	2750.0006	1297.58	620	2750	2750.0006	40.77	0
16	8	10^{-6}	1	660	2450	2450.0007	936.54	670	2450	2450.0007	45.81	0
17	7	10^{-6}	1	750	2150	2150.0008	881.24	770	2150	2150.0008	47.78	0
18	6	10^{-6}	1	880	1850	1850.0009	780.45	900	1900	1900.0009	42.71	2.70
19	5	10^{-6}	1	1050	1550	1550.0011	669.91	1070	1550	1550.0011	48.56	0
20	4	10^{-6}	1	1300	1250	1250.0013	573.69	1330	1300	1300.0013	45.81	4.00
21	3	10^{-6}	1	1750	950	950.0018	480.12	1790	950	950.0018	44.69	0
22	2	10^{-6}	1	2560	650	650.0026	378.07	2600	650	650.0026	46.99	0

For the NSGA-II, the population size (N) is set to 30, the maximum generation (G_{max}) is set to 300, the crossover rate (P_c) is set to 0.85, and the mutation rate (P_m) is set to 0.01. The result for K=5 is shown in Table 10. The computational time is 62.16 seconds. Fig. 14 shows the convergence of the NSGA-II with K=5 under three dimensions. Fig. 15 shows the convergence of the NSGA-II under two dimensions. There are five Pareto optimal assignments (CT, CH): (750, 1750), (780, 1700), (870, 1650), (1000, 1600) and (1150, 1550). For the MIP, as shown in Table 9, both scenarios 8 (w_1 = 1 and w_2 = 10^{-6}) and 19 (w_1 = 10^{-6} and w_2 = 1) are for K=5, and the optimal solution is (720, 1750) and (1050, 1550), respectively. With no weights given to the two objectives under the NSGA-II, there are five solutions on the Pareto frontier. The two optimal solutions under the MIP are not exactly the same under the NSGA-II. For (720, 1750) under the MIP, the solution is (750, 1750) under the NSGA-II, and (730, 1750) under the MOGA. For (1050, 1550) under the MIP, the solution is (1150, 1550) under the NSGA-II, and (1070, 1550) under the MOGA. The additional solutions under the NSGA-II are (780, 1700), (870, 1650) and (1000, 1600).

Table 10
Results for Case 3 with K=5 using NSGA-II

K	(CT, CH)	CPU(s)
5	(750, 1750) (780, 1700) (8701, 1650) (100, 1600) (1150,1550)	62.16

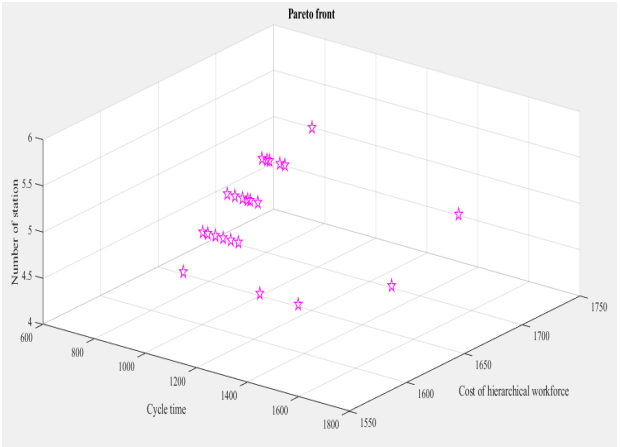


Fig. 14. The convergence of NSGA-II with K=5 for Case 3 under 3D

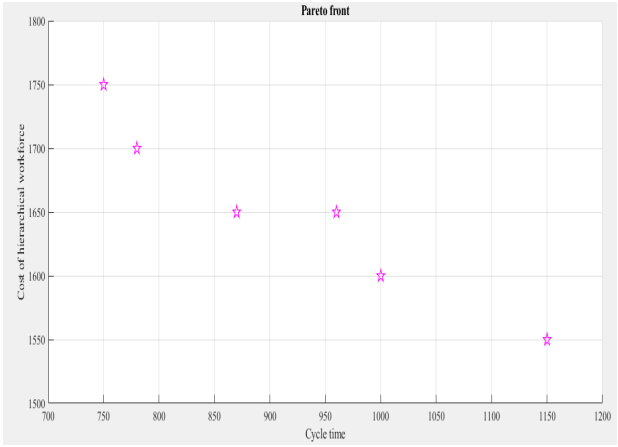


Fig. 15. The convergence of NSGA-II with K=5 for Case 3 under 2D

4.4 Case 4

The precedence relationship for Case 4 is from Tonge (1961) and Li et al. (2018). Since the past works did not consider mixed models and hierarchical workers, we need to provide some additional information. In Case 4, three product models are produced. There are a total of 70 tasks. Three types of workers are available: type I (highly skilled), type II (skilled), and type III (normal). The unit costs for type I, II, and III workers are \$350, \$300, and \$250, respectively. Table A2 in the Appendix shows the task information of the 70 tasks, including the immediate predecessor(s) of each task and the processing time of each task for each model by each type of worker. The calculated upper bound of stations (*UB*) is 24. Therefore, let the number of stations (*K*) range from 2 to 24. The MIP model can no longer solve the problem. For the MOGA, the population size (*N*), crossover rate (P_c), mutation rate (P_m), and stop generation (G_{max}) are set to 50, 0.85, 0.05, and 500, respectively. The process of each scenario is repeated three times, and the results with the best objective value are collected. The solutions under the proposed MOGA are shown in Table 11. For instance, both scenarios 11 and 26 have five stations (K =5). With weights $w_1 = 1$ and $w_2 = 10^{-6}$ in scenario 11, the MOGA has a solution with a cycle time (*CT*) of 2670 seconds and a total cost of the hierarchical workforce (*CH*) of \$2100, and the objective value is 2670.0021. The computational time for the MOGA is 53.32 seconds. With weights $w_1 = 10^{-6}$ and $w_2 = 1$ in scenario 26, the MOGA has a solution with a cycle time (*CT*) of 4860 seconds and a total cost of the hierarchical workforce (*CH*) of \$1,650, and the objective value is 1650.0049. The computational time for the MOGA is 55.65 seconds. For the NSGA-II, the population size (*N*) is set to 30, the maximum generation (G_{max}) is set to 300, the crossover rate (P_c) is set to 0.85, and the mutation rate (P_m) is set to 0.01. Table 12 shows the result for K=6. The computational time is 87.86 seconds. Fig. 16 depicts the convergence of the NSGA-II with K=6 under three dimensions, i.e. cycle time, total cost of hierarchical workforce, and number of stations. Fig. 17 depicts the convergence of the NSGA-II under two dimensions. There are eight Pareto optimal assignments, as listed in Table 12. Note that while two solutions are found for K=6 in the MOGA (scenarios 11 and 26), eight solutions are found in the NSGA-II. The two solutions found in the MOGA are not the same as those found in the NSGA-II

Table 11
Results for Case 4 by the MOGA

Scenario	*K*	w_1	w_2	*CT*	*CH*	*f*	CPU(s)
1	16	1	10^{-6}	1080	5600	1080.0056	58.23
2	15	1	10^{-6}	1080	5250	1080.0053	55.12
3	14	1	10^{-6}	1170	4900	1170.0049	56.65
4	13	1	10^{-6}	1260	4550	1260.0046	57.52
5	12	1	10^{-6}	1350	4200	1350.0042	56.98
6	11	1	10^{-6}	1470	3850	1470.0039	51.36
7	10	1	10^{-6}	1620	3500	1620.0035	52.23
8	9	1	10^{-6}	1770	3150	1770.0032	53.12
9	8	1	10^{-6}	2010	2800	2010.0028	56.01
10	7	1	10^{-6}	2280	2450	2280.0025	56.36
11	6	1	10^{-6}	2670	2100	2670.0021	53.32
12	5	1	10^{-6}	3180	1750	3180.0018	56.25
13	4	1	10^{-6}	3990	1400	3990.0014	55.54
14	3	1	10^{-6}	5310	1050	5310.0011	51.96
15	2	1	10^{-6}	7950	700	7950.0007	52.91
16	16	10^{-6}	1	2160	4150	4150.0022	58.05
17	15	10^{-6}	1	2160	3900	3900.0022	55.78
18	14	10^{-6}	1	2160	3650	3650.0022	56.69
19	13	10^{-6}	1	2360	3400	3400.0024	57.15
20	12	10^{-6}	1	2520	3150	3150.0025	53.09
21	11	10^{-6}	1	2700	2900	2900.0027	58.59
22	10	10^{-6}	1	3000	2650	2650.0030	56.36
23	9	10^{-6}	1	3300	2400	2400.0033	59.45
24	8	10^{-6}	1	3660	2150	2150.0037	56.07
25	7	10^{-6}	1	4200	1900	1900.0042	61.25
26	6	10^{-6}	1	4860	1650	1650.0049	55.65
27	5	10^{-6}	1	5880	1350	1350.0059	58.15
28	4	10^{-6}	1	7320	1100	1100.0073	51.45
29	3	10^{-6}	1	9720	850	850.0097	58.36
30	2	10^{-6}	1	14580	600	600.0146	59.45

Table 12
Results for Case 4 using NSGA-II

K	(*CT*, *CH*)	CPU(s)
6	(2740, 2100) (2860, 2000) (3040, 1950) (3220, 1900) (3435, 1800) (4120, 1750) (4360, 1700) (5200, 1650)	87.86

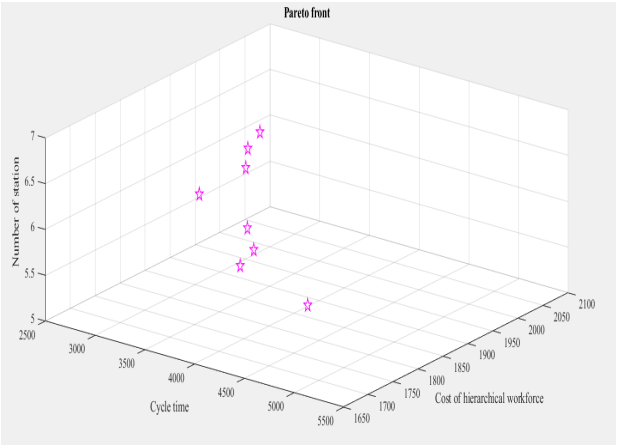


Fig. 16. The convergence of NSGA-II with K=6 for Case 4 under 3D

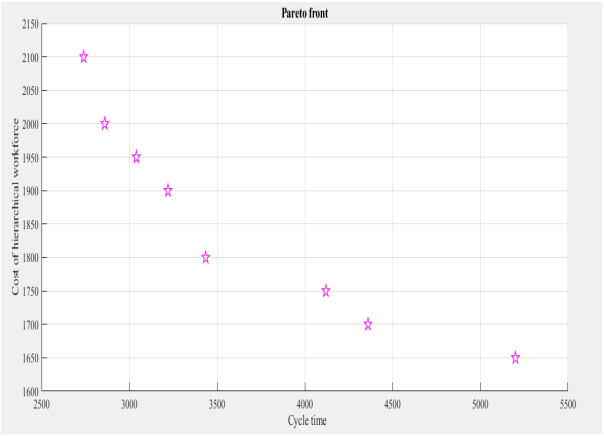


Fig. 17. The convergence of NSGA-II with K=5 for Case 4 under 2D

4.5 *Case 5*

In Case 5, four product models are produced, and the data is obtained and revised from Bartholdi (1993), Özcan and Toklu (2009) and Campana et al. (2022). Parameter ρ is the time factor that defines the time scaling from worker type h to worker type h+1, used in such a way that $t_{jm}^{h+1} = \rho t_{jm}^{h}$. In Case 5, ρ is set to 1.2. There are 148 tasks and three types of workers, and the unit costs for type I, II, and III workers are \$350, \$300, and \$250, respectively.

Table 13
Results for Case 5 by the proposed MOGA

Scenario	K	CT	CH	f	CPU(s)	Scenario	K	CT	CH	f	CPU(s)
1	32	1310	11200	1310.0112	194.05	32	32	2060	8150	8150.0021	191.11
2	31	1310	10850	1310.0109	189.01	33	31	2060	7900	7900.0021	195.81
3	30	1310	10500	1310.0105	195.81	34	30	2060	7650	7650.0021	195.92
4	29	1310	10150	1310.0102	200.11	35	29	2060	7400	7400.0021	202.03
5	28	1310	9800	1310.0098	189.55	36	28	2060	7150	7150.0021	187.89
6	27	1310	9450	1310.0095	193.59	37	27	2060	6900	6900.0021	188.95
7	26	1310	9100	1310.0091	195.89	38	26	2060	6650	6650.0021	185.24
8	25	1310	8750	1310.0088	196.81	39	25	2060	6400	6400.0021	196.59
9	24	1320	8400	1320.0084	189.69	40	24	2060	6150	6150.0021	198.91
10	23	1380	8050	1380.0081	185.78	41	23	2060	5900	5900.0021	199.74
11	22	1410	7700	1410.0077	198.59	42	22	2200	5650	5650.0022	185.88
12	21	1460	7350	1460.0074	196.33	43	21	2260	5400	5400.0023	196.86
13	20	1530	7000	1530.0070	195.19	44	20	2320	5150	5150.0023	199.76
14	19	1610	6650	1610.0067	197.57	45	19	2380	4900	4900.0024	193.19
15	18	1710	6300	1710.0063	186.91	46	18	2490	4650	4650.0025	184.95
16	17	1800	5950	1800.0060	192.19	47	17	2520	4400	4400.0025	206.91
17	16	1850	5600	1850.0056	187.96	48	16	2690	4150	4150.0027	197.93
18	15	1990	5260	1990.0053	205.23	49	15	3020	3850	3850.0030	198.84
19	14	2160	4900	2160.0049	187.88	50	14	3080	3600	3600.0031	189.71
20	13	2290	4550	2290.0046	188.91	51	13	3240	3350	3350.0032	188.76
21	12	2520	4200	2520.0042	185.33	52	12	3610	3100	3100.0036	185.91
22	11	2610	3850	2610.0039	193.59	53	11	3810	2850	2850.0038	188.83
23	10	2890	3500	2890.0035	198.51	54	10	4090	2600	2600.0041	189.92
24	9	3240	3150	3240.0032	199.15	55	9	4460	2350	2350.0045	185.94
25	8	3570	2800	3570.0028	185.99	56	8	4980	2100	2100.0050	194.54
26	7	4040	2450	4040.0025	189.84	57	7	5640	1850	1850.0056	198.16
27	6	4700	2100	4700.0021	195.73	58	6	6570	1600	1600.0066	199.19
28	5	5640	1750	5640.0018	193.22	59	5	7870	1350	1350.0079	186.91
29	4	7040	1400	7040.0014	185.91	60	4	9840	1100	1100.0098	189.17
30	3	9380	1050	9380.0011	202.02	61	3	13110	850	850.0131	189.94
31	2	14070	700	14070.0007	198.59	62	2	19660	600	600.0197	189.56

Table A3 in the Appendix shows the task information of the 148 tasks, including the immediate predecessor(s) of each task. Tasks 1 to 10 can only be processed by type I workers; tasks 11 to 20 can only be processed by either type I or type II workers; and all types of workers can process the rest of the tasks. The calculated upper bound of stations (UB) is 32, and the number of stations (K) ranges from 2 to 32. As in Case 4, the MIP model can no longer solve the problem. For the MOGA, the population size (N), crossover rate (P_c), mutation rate (P_m), and stop generation (G_{max}) are set to 50, 0.88, 0.06, and 500, respectively. Under scenarios 1 to 31, we let w_1=1 and w_2=10^{-6}. Under scenario 32 to 62, we let w_1=10^{-6} and w_2=1. The process of each scenario is repeated three times, and the results with the best objective value are collected. The solutions

under the proposed MOGA are shown in Table 13. For example, both scenarios 23 and 54 have ten stations (K =10). With weights $w_1 = 1$ and $w_2 = 10^{-6}$ in scenario 23, the MOGA has a solution with a cycle time (CT) of 2890 seconds and a total cost of the hierarchical workforce (CH) of \$3500, and the objective value is 2890.0035. The computational time for the MOGA is 198.51 seconds. With weights $w_1 = 10^{-6}$ and $w_2 = 1$ in scenario 54, the MOGA has a solution with a cycle time (CT) of 4090 seconds and a total cost of the hierarchical workforce (CH) of \$2600, and the objective value is 2600.0041. The computational time for the MOGA is 189.92 seconds.

For the NSGA-II, the population size (N), the maximum generation (G_{max}), the crossover rate (P_c), and the mutation rate (P_m) are set to 30, 300, 0.85, and 0.01, respectively. Table 14 shows the result for K=10. The computational time is 656.82 seconds. Fig. 18 depicts the convergence of the NSGA-II with K=10 under three dimensions. Fig. 19 depicts the convergence of the NSGA-II under two dimensions. There are 12 Pareto optimal assignments, as listed in Table 14. Note that the two solutions found in the MOGA are not the same as those found in the NSGA-II.

Table 14
Results for Case 5 using NSGA-II

K	(CT, CH)	CPU(s)
10	(2930, 3500) (3010, 3450) (3090, 3400) (3240, 3250) (3590, 3150) (3650, 3100) (3780, 3050) (3890, 2950) (3940, 2900) (4010, 2850) (4150, 2750) (4360, 2600)	656.82

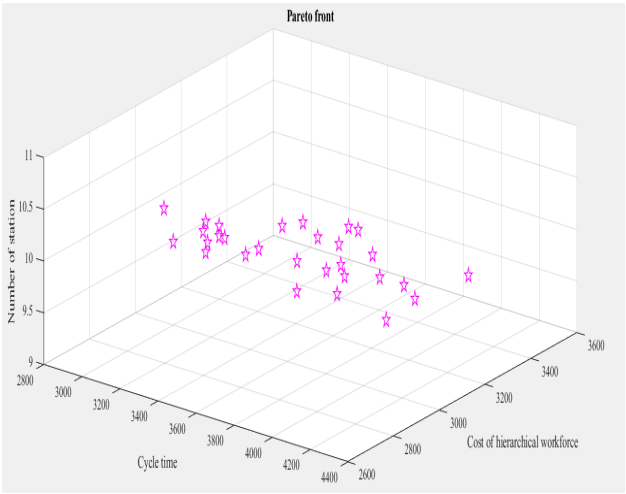


Fig. 18 The convergence of NSGA-II with K=10 for Case 5 under 3D

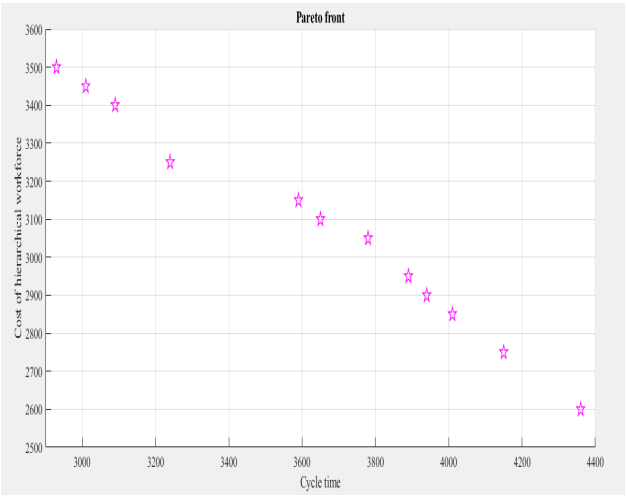


Fig. 19 The convergence of NSGA-II with K=10 for Case 5 under 2D

4.6 Comparisons of models

Five cases are used to examine the proposed MIP model, the MOGA model and the NSGA-II model for the MO-MALBP-HW. Due to the small problem scale of Case 1, the three models can obtain optimal solutions in a short calculation time. The computational time required by the MOGA is higher than that required by the MIP because many generations are required before termination is reached. In addition, the NSGA-II requires higher computational time than the MOGA because the NSGA-II needs to obtain all Pareto optimal solutions. In Case 2, the MIP can obtain the optimal solutions. While the MOGA can generate optimal solutions for some scenarios, near-optimal solutions are obtained for other scenarios, with %Gap of zero. The NSGA-II can obtain the two optimal solutions generated by the MIP, and also some additional Pareto solutions. With a larger scale in Case 3, the MOGA and the NSGA-II can still generate near-optimal solutions under a short computational time. The computational times required by the MOGA and the NSGA-II under different scenarios are relatively more stable than those by the MIP. Compared with the first three cases, the scales of Cases 4 and 5 are relatively large, and the MIP cannot generate the optimal solutions anymore. However, the MOGA model and the NSGA-II can still obtain satisfactory solutions quickly. In summary, the MIP can obtain optimal solutions when the problem size is small, while the MOGA model and the NSGA-II can efficiently obtain satisfactory solutions for large instances.

Three related works, Simaria and Vilarinho (2004), Saif et al. (2014), and Campana et al. (2022), are compared with the proposed MIP model, the MOGA model and the NSGA-II model. Table 15 shows that the MIP, the MOGA model, and the NSGA-II model consider more features than the other three works. Since the MOGA model and the NSGA-II can obtain satisfactory solutions in a short time, they can be applied to solve large-scale problems.

Table 15
Comparisons among related works

Compared items	Simaria and Vilarinho (2004)	Saif *et al.* (2014)	Campana *et al.* (2022)	Proposed MIP	Proposed MOGA	Proposed NSGA-II
Algorithm	Genetic algorithm	Artificial bee colony algorithm	Variable neighborhood descent algorithm	Exact	Genetic algorithm	Genetic algorithm
Accuracy	Near-optimal	Near-optimal	Near-optimal	Optimal	Near-optimal	Near-optimal
ALBP	√	√	√	√	√	√
Mixed-model	√	√		√	√	√
Non-dominated Sorting						√
Positive zoning				√	√	√
Negative zoning				√	√	√
Hierarchical workers			√	√	√	√
Multiple objectives	√	√		√	√	√
Number of workstations	√	√	√	√	√	√
Cost of hierarchical workers			√	√	√	√
Cycle time	√	√	√	√	√	√
Solve large-scale problem	√	√	√		√	√
Consider binary behavior	√	√	√	√	√	√
Solve by free software packages				√		

5. Conclusions

In response to growing customer demand of diversified products, manufacturers have begun manufacturing different models with varying properties on an assembly line. As a result, mixed-model assembly lines have emerged and are widely used in many industries. The mixed-model assembly line balancing problem (MALBP) has attracted the attention of many scholars in recent years, and it is basically an NP-hard problem. Including hierarchical worker assignment makes the problem even more difficult to solve. Since processing requirements for tasks often vary and workers may have different qualification levels, the multi-objective mixed-model assembly line balancing problem with hierarchical worker assignment (MO-MALBP-HW) is present in today's manufacturing environment.

This research proposed a mixed integer programming (MIP) model, a multi-objective genetic algorithm (MOGA) approach, and a non-dominated sorting genetic algorithm II (NSGA-II) to solve the MO-MALBP-HW. The MIP model can solve small-scale problems efficiently and optimally. When the scale of the problem increases, the MIP model can no longer obtain the solution. However, the MOGA model and the NSGA-II model can still obtain near-optimal solutions to large-scale problems in a short computational time. In addition, the MOGA model can consider different weights of the objectives, and the NSGA-II can obtain Pareto optimal solutions without considering the weights of the objectives. The proposed models can help develop the multi-objective mixed-model assembly line balancing plan and improve plant production efficiency. To the best of the authors' knowledge, this is the first work that simultaneously considers multiple objectives, mixed-model assembly line balancing, and hierarchical worker assignment. Thus, the proposed models can tackle a more realistic problem in practice.

With the rapid development of technology and the manufacturing environment, various new assembly line balancing problems have emerged. For example, a multi-objective mixed-model two-sided assembly line balancing problem with hierarchical worker assignment can be studied, or the problem with using robots and green manufacturing can be considered. New methods and models can be developed to solve real-world problems and help manufacturers perform better.

Acknowledgements

This work was partly supported by the National Science and Technology Council in Taiwan under Grant NSTC-112-2637-E-167-001. The authors would like to thank Dr. Tzuoh-Yi Lin, manager at Li Ming Machinery Co., Ltd., for his insights into and data on the problem.

References

Ab. Rashid, M.F.F., Mohd Rose, A.N., Nik Mohamed, N.M.Z., & Mohd Romlay, F.R. (2020). Improved moth flame optimization algorithm to optimize cost-oriented two-sided assembly line balancing. *Engineering Computations, 37*(2), 638-663.

Alhomaidhi, E. (2024). Enhancing efficiency and adaptability in mixed model line balancing through the fusion of learning effects and worker prerequisites. *International Journal of Industrial Engineering Computations*, *15*(2), 541-552.

Bartholdi, J.J. (1993). Balancing two-sided assembly lines: a case study. *International Journal of Production Research, 31*(10), 2447-2461.

Belkharroubi, L., & Yahyaoui, K. (2022). Solving the mixed-model assembly line balancing problem type-I using a hybrid reactive GRASP. *Production and Manufacturing Research, 10*(1), 108-131.

Campana, N.P., Iori, M., & Moreira, M.C.O. (2022). Mathematical models and heuristic methods for the assembly line balancing problem with hierarchical worker assignment. *International Journal of Production Research, 60*(7), 2193-2211.

Chen, J.C., Chen, Y.Y., Chen, T.L., & Kuo, Y.H. (2019). Applying two-phase adaptive genetic algorithm to solve multi-model assembly line balancing problems in TFT-LCD Module Process. *Journal of Manufacturing Systems, 52*, 86-99.

Chen, J., Jia, X., & He, Q. (2023). A novel bi-level multi-objective genetic algorithm for integrated assembly line balancing and part feeding problem. *International Journal of Production Research, 61*(2), 580-603.

Chutima, P., & Khotsaenlee, A. (2022). Multi-objective parallel adjacent U-shaped assembly line balancing collaborated by robots and normal and disabled workers. *Computers & Operations Research*, *143*, 105775.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197.

Delice, Y., Aydoğan, E.K., Himmetoğlu, S., & Özcan, U. (2023). Integrated mixed-model assembly line balancing and parts feeding with supermarkets. *CIRP Journal of Manufacturing Science and Technology, 41*, 1-18.

Faccio, M., Gamberi, M., & Bortolini, M. (2016). Hierarchical approach for paced mixed-model assembly line balancing and sequencing with jolly operators. *International Journal of Production Research, 54*(3), 761-777.

Goldberg, D.E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems, 3*(5), 493-530.

Kang, H.-Y., & Lee, A.H.I. (2023). An evolutionary genetic algorithm for a multi-objective two-sided assembly line balancing problem: a case study of automotive manufacturing operations. *Quality Technology and Quantitative Management, 20*(1), 66-88.

Korytkowski, P. (2017). Competences-based performance model of multi-skilled workers with learning and forgetting. *Expert Systems with Applications, 77*, 226-235.

Lee, A.H. I., Kang, H.-Y., & Chen, C.-L. (2021). Multi-objective assembly line balancing problem with setup times using fuzzy goal programming and genetic algorithm. *Symmetry, 13*(2), 1-28.

Li, D., Zhang, C., Tian, G., Shao, X., & Li, Z. (2018). Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48*(1), 119-129.

Li, S., Butterfield, J., & Murphy, A. (2023). A new multi-objective genetic algorithm for assembly line balancing. Journal of Computing and Information Science in Engineering, *23*(3), 034502.

Liao, S.-G., Zhang, Y.-B., Sang, C.-Y., & Liu, H. (2023). A genetic algorithm for balancing and sequencing of mixed-model two-sided assembly line with unpaced synchronous transfer. *Applied Soft Computing, 146*, 110638.

Lin, W.Y., Lee, W.Y., & Hong, T.P. (2003). Adapting crossover and mutation rates in genetic algorithms. *Journal of Information Science and Engineering,* 19(5), 889-903.

LINGO User's Manual (2018). version 18. LINGO System Inc., Chicago.

Liu, X., Yang, X., & Lei, M. (2021). Optimisation of mixed-model assembly line balancing problem under uncertain demand. *Journal of Manufacturing Systems, 59*, 214-227.

Liu, Y., Shen, W., Zhang, C., & Sun, X. (2023). Agent-based simulation and optimization of hybrid flow shop considering multi-skilled workers and fatigue factors. *Robotics and Computer-Integrated Manufacturing, 80*, 102478.

Małachowski, B., & Korytkowski, P. (2016). Competence-based performance model of multi-skilled Workers. *Computers and Industrial Engineering, 91*, 165–177.

MATLAB User's Manual (2019). version 2019. The MathWorks, Inc., Massachusetts.

Meng, K., Tang, Q., Cheng, L., & Zhang, Z. (2022). Mixed-model assembly line balancing problem considering preventive maintenance scenarios: MILP model and cooperative co-evolutionary algorithm. *Applied Soft Computing, 127*, 109341.

Nourmohammadi, A., Ng, A.H.C., Fathi, M., Vollebregt, J., Hanson, L. (2023). Multi-objective optimization of mixed-model assembly lines incorporating musculoskeletal risks assessment using digital human modeling. *CIRP Journal of Manufacturing Science and Technology*, *47*, 71-85.

Özcan, U., & Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. *Computers and Industrial Engineering, 57*, 217-227.

Rabbani, M., Behbahan, S.Z.B., & Farrokhi-Asl, H. (2020). The collaboration of human-robot in mixed-model four-sided assembly line balancing problem. *Journal of Intelligent & Robotic Systems, 100*, 71-81.

Rahman, H.F., Janardhanan, M.N., & Ponnambalam, S.G. (2023). Energy aware semi-automatic assembly line balancing problem considering ergonomic risk and uncertain processing time. *Expert Systems with Applications, 231*, 120737.

Razali, M.M., Kamarudin, N.H., Ab. Rashid, M.F.F., & Mohd Rose, A.N. (2019). Recent trend in mixed-model assembly line balancing optimization using soft computing approaches. *Engineering Computations, 36*(2), 622-645.

Saif, U., Guan, Z., Liu, W. Wang, B., & Zhang, C. (2014). Multi-objective artificial bee colony algorithm for simultaneous sequencing and balancing of mixed model assembly line. *International Journal of Advanced Manufacturing Technology, 75*, 1809-1827.

Samouei, P., & Sobhishoja, M. (2023). Robust counterpart mathematical models for balancing, sequencing, and assignment of robotic U-shaped assembly lines with considering failures and setup times. *OPSEARCH*, *60*, 87-124.

Simaria, A.S., & Vilarinho, P.M. (2004). A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. *Computers & Industrial Engineering, 47*, 391-407.

Soysal-Kurt, H., İşleyen, S.K., Gökçen, H. (2024). Balancing and sequencing of mixed-model parallel robotic assembly lines considering energy consumption. *Flexible Services and Manufacturing Journal,* Online first.

Sun, X., Guo, S., Guo, J., Du, B., Yang, Z., & Wang, K. (2024). A Pareto-based hybrid genetic simulated annealing algorithm for multi-objective hybrid production line balancing problem considering disassembly and assembly. *International Journal of Production Research, 62*(13):4809-4830.

Sungur, B., & Yavuz, Y. (2015). Assembly line balancing with hierarchical worker assignment. *Journal of Manufacturing Systems, 37*, 290-298.

Tanhaie, F., Rabbani, M., & Manavizadeh, N. (2020). Simultaneous balancing and worker assignment problem for mixed-model assembly lines in a make-to-order environment considering control points and assignment restrictions. *Journal of Modelling in Management, 15*(1), 1-34.

Tonge, F. M. (1961). *A heuristic program for assembly line balancing*. Prentice-Hall, New Jersey.

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. *Journal of Cleaner Production, 244*, 118845.

Zhang, J.H., Li, A.P., & Liu, X.M. (2019). Hybrid genetic algorithm for a type-II robust mixed-model assembly line balancing problem with interval task times. *Advances in Manufacturing, 7*, 117-132.

Zhang, Z., Tang, Q., & Chica, M. (2021). A robust MILP and gene expression programming based on heuristic rules for mixed-model multi-manned assembly line balancing. *Applied Soft Computing, 109*, 107513.

Appendix

Table A1

Task information for Case 3

Task	Predecessors	Type I worker			Type II worker		
		Model A	Model B	Model C	Model A	Model B	Model C
1		90	60	60	+∞	+∞	+∞
2	1	90	60	90	+∞	+∞	+∞
3	1	90	60	60	+∞	+∞	+∞
4	2	0	60	60	+∞	+∞	+∞
5	3	150	120	0	+∞	+∞	+∞
6	4	150	120	150	+∞	+∞	+∞
7	1	120	90	120	+∞	+∞	+∞
8	2	120	90	120	+∞	+∞	+∞
9	5,7	150	120	150	+∞	+∞	+∞
10	6,8	150	120	120	+∞	+∞	+∞
11		90	60	60	150	100	100
12		90	60	90	150	100	150
13	11,12	30	0	30	50	0	50
14	7,8,13	150	120	120	250	200	200
15	13	90	60	90	150	100	150
16	15	90	120	120	150	200	200
17	14	30	60	30	50	100	50
18	15	60	0	60	100	0	100
19	16,18	30	60	0	50	100	0
20	19	0	30	30	0	50	50
21	20	300	330	300	500	550	500
22	21	60	90	90	100	150	150
23	15	150	180	180	250	300	300
24	15	150	180	150	250	300	250
25	14	120	150	150	200	250	250
26	17,25	0	30	30	0	50	50
27	17	0	30	30	0	50	50
28	22,27	120	150	120	200	250	200
29	14	30	60	30	50	100	50
30	14	0	30	0	0	50	0
31	14	60	60	60	100	100	100
32	14	30	0	30	50	0	50
33	19,23,24,27	90	90	90	150	150	150
34	33	30	30	30	50	50	50
35	33	60	60	60	100	100	100
36	33	60	60	60	100	100	100
37	12	90	0	90	150	0	150
38	26,28,34,36	60	60	0	100	100	0
39		30	30	30	50	50	50
40	35,38	90	60	60	150	100	100
41	9,10,29,30,31,32,39,40	120	120	120	200	200	200
42	41	90	60	60	150	100	100

Table A2
Task information for Case 4

Task	Predecessor	Type I worker			Type II worker			Type III worker		
		Model A	Model B	Model C	Model A	Model B	Model C	Model A	Model B	Model C
1		150	90	120	+∞	+∞	+∞	+∞	+∞	+∞
2	1	420	360	420	+∞	+∞	+∞	+∞	+∞	+∞
3	2	360	300	330	+∞	+∞	+∞	+∞	+∞	+∞
4	3	330	270	300	+∞	+∞	+∞	+∞	+∞	+∞
5		60	0	0	+∞	0	0	+∞	0	0
6	4,5	390	360	360	+∞	+∞	+∞	+∞	+∞	+∞
7	4	120	90	120	+∞	+∞	+∞	+∞	+∞	+∞
8	6,7	150	90	120	+∞	+∞	+∞	+∞	+∞	+∞
9		120	90	120	+∞	+∞	+∞	+∞	+∞	+∞
10	9	60	60	60	+∞	+∞	+∞	+∞	+∞	+∞
11	10	120	90	120	200	150	200	+∞	+∞	+∞
12	8,11	210	150	180	350	250	300	+∞	+∞	+∞
13	12	180	180	150	300	300	250	+∞	+∞	+∞
14	12	60	60	60	100	100	100	+∞	+∞	+∞
15		90	90	90	150	150	150	+∞	+∞	+∞
16	15	150	150	120	250	250	200	+∞	+∞	+∞
17	16	120	120	90	200	200	150	+∞	+∞	+∞
18	16	90	60	90	150	100	150	+∞	+∞	+∞
19	17,18	150	90	120	250	150	200	+∞	+∞	+∞
20	19	120	120	90	200	200	150	+∞	+∞	+∞
21	20	330	270	330	550	450	550	660	540	660
22	19	270	210	270	450	350	450	540	420	540
23	13,14,21,22	480	420	480	800	700	800	960	840	960
24	5	120	60	120	20	10	200	240	120	240
25	23,24	930	870	930	155	145	1550	1860	1740	1860
26	25	210	240	240	35	40	400	420	480	480
27	25	240	270	270	40	45	450	480	540	540
28	25	420	450	450	70	75	750	840	900	900
29	25	120	150	150	20	25	250	240	300	300
30	5	30	60	60	5	10	100	60	120	120
31	23,30	150	180	180	25	30	300	300	360	360
32	31	270	300	300	45	50	500	540	600	600
33	23	600	630	600	100	105	1000	1200	1260	1200
34	33	240	270	240	40	45	400	480	540	480
35	26,27,28,	180	210	300	300	350	300	360	420	360
36	35	210	240	210	350	400	350	420	480	420
37	36	0	30	30	0	50	50	0	60	60
38	37	0	30	30	0	50	50	0	60	60
39	38	0	30	60	0	50	100	0	60	120
40	39	60	90	0	100	150	0	120	180	0
41	1	90	120	0	150	200	0	180	240	0
42	40,41	120	150	120	200	250	200	240	300	240
43	42	90	120	90	150	200	150	180	240	180
44	35	240	270	240	400	450	400	480	540	480
45	44	150	180	150	250	300	250	300	360	300
46	45	480	510	480	800	850	800	960	1020	960
47	46	510	540	510	850	900	850	1020	1080	1020
48	35	300	330	300	500	550	500	600	660	600
49	48	330	360	330	550	600	550	660	720	660
50	43,47,49	240	270	240	400	450	400	480	540	480
51	35	60	0	60	100	0	100	120	0	120
52	51	150	90	0	250	150	0	300	180	0
53	35	270	210	240	450	350	400	540	420	480
54	52,53	720	660	690	1200	1100	1150	1440	1320	1380
55	54	240	180	210	400	300	350	480	360	420
56	35	420	360	390	700	600	650	840	720	780
57	19	120	60	90	200	100	150	240	120	180
58	57	60	0	60	100	0	100	120	0	120
59	58	90	30	90	150	50	150	180	60	180
60	35,59	180	120	0	300	200	0	360	240	0
61	35	150	90	120	250	150	200	300	180	240
62	35	180	120	180	300	200	300	360	240	360
63	62	1080	1020	990	1800	1700	1650	2160	2040	1980
64	63	180	120	150	300	200	250	360	240	300
65	61,64	90	30	60	150	50	100	180	60	120
66	64	150	90	120	250	150	200	300	180	240
67	64	120	60	90	200	100	150	240	120	180
68	3	450	390	450	750	650	750	900	780	900
69	1	150	90	150	250	150	250	300	180	300
70	1	180	120	150	300	200	250	360	240	300

Table A3
Task information for Case 5

Task	Successors	Task	Successors	Task	Successors	Task	Successors
1	5,6,7,8	38	39	75	88,97	112	113
2	3	39	40	76	77	113	114,116,120,123,128
3	4,5,6,7	40	41,48,55	77	78	114	115
4	8	41		78	79	115	125
5	14	42	43	79	80	116	117
6	9	43	44	80	81	117	118
7	14	44		81	106	118	126
8	10	45	46	82	83,89,143,146	119	
9	14	46	47	83		120	121
10	14	47	48,49,55	84	85	121	122
11	12	48		85		122	126
12	13	49		86		123	124
13		50	51	87		124	125
14	15,16	51	53,69	88	111	125	
15	17	52	53	89	90	126	
16	17	53		90	79	127	
17	18,19	54	133	91	105	128	129
18	20	55	54,72,76,87,88	92	135	129	130
19	20	56	73	93		130	131,137
20	21,22,23,24	57	79	94		131	
21	25,26,27,28	58	84,86	95	101	132	135
22	25,26,27,28	59	75,87	96	104	133	135
23	25,26,27,28	60		97		134	135
24	25,26,27,28	61	62	98	101	135	136
25	29	62	63	99	100	136	
26	29	63	67	100	101	137	
27	29	64	65,71,72	101	102,103	138	139
28	29	65	66,99	102	127	139	140
29	31	66	67	103	127	140	
30		67	68	104		141	142
31	36	68	95,98	105	119	142	143,146,147,148
32	34	69	82	106	107	143	
33	35	70	71	107	108	144	145
34	36	71		108	109	145	147,148
35	36	72	134	109	110	146	
36	37	73	84,86,87,88,96	110		147	
37	38,45	74	75	111	112	148