Journal of Industrial and Management Optimization

JIMO, 22(3): 1325–1360.
DOI: 10.3934/jimo.2026049
Received: 03 December 2025
Revised: 26 January 2026
Accepted: 06 February 2026
Published: 11 February 2026

https://www.aimspress.com/journal/jimo

Research article

A mathematical model and hyperheuristic algorithm for integrated cell formation, cell layout and scheduling problems with a material-handling robot constraints

Gulcin Bektur[1,*] and Mehmet Palta[2]

[1] Department of Industrial Engineering, Hitit University, Corum 19169, Turkiye
[2] Department of Industrial Engineering, Iskenderun Technical University, Hatay 31200, Turkiye

* **Correspondence:** Email: gulcinbektur@hitit.edu.tr; Tel: +90-505-490-4151.

Abstract: To increase the efficiency of cellular manufacturing systems (CMSs), decision problems that affect each other should be addressed in an integrated manner. In this study, integrated cell formation (CF), cell layout (CL), and part scheduling problems are addressed by considering the material handling system and alternative routes. The job shop production environment is considered, and material-handling robots are used for intercell and intracell part transportation between machines. It is assumed that the machines are of different sizes. Additionally, a mathematical model is proposed. By solving the proposed mathematical model, decisions are made to assign parts to cells, to determine the appropriate route for each part, the layout of cells and machines, and the transportation order of the parts on the material-handling robots, and to schedule the parts on the machines in accordance with their routes. A hyperheuristic (HH) algorithm is also proposed, and the success of this algorithm is demonstrated on test problems.

Keywords: cell formation and cell layout; job shop scheduling; material-handling robot constraint; heuristic algorithm; mathematical model
Mathematics Subject Classification: 90C59, 90C11, 90C27

1. Introduction

Group technology (GT) is a production philosophy that involves the economic production of different types of parts by exploiting similarities in their production processes. By grouping different machines in a cell, part families with similar production processes can be produced economically within cells [1].

CMSs are GT-based production systems. One of the decision problems involved in CMSs is the CF problem. In the CF problem, machines are assigned to cells, and part families are constructed [2]. Each part has a route and should be processed on the machines after the route is considered. If a part is processed on machines in different cells, the part is transported between cells; if that part is processed on machines in the same cell, it is transported between the machines in the cell. Considering the GT philosophy, it is logical to produce parts with similar production processes in the same cell to reduce material handling between cells [3]. However, because current manufacturing conditions involve product mixes with very different production processes, it may not be possible to obtain independent cells. In this case, parts are transported not only within the cell but also between cells.

Since the transportation of parts does not add value to the production process, high amounts of transportation result in inefficient production processes. Furthermore, transportation times are significantly affected by the location of the machines and the cell; thus, in some studies, the CF and cell layout (CL) problems are considered together. Regarding the CL problem, the location of the cells and the location of the machines in each cell can be determined [4,5], and the material handling time of the parts affects their completion time. Therefore, CF, CL, and scheduling activities are important decision problems that are affected by each other [2]. By solving the part scheduling problem, the part completion time can be determined, as it is important to address these problems in an integrated manner to obtain effective solutions instead of solving them separately.

Material handling systems are important elements that affect production processes. Today, there are many alternatives for material handling systems, and as a result of increasing technological opportunities, businesses have started to use material-handling robots for transportation. The use of material-handling robots makes it possible to achieve energy savings and faster material transportation. However, the material handling system has been neglected in many studies.

In this study, robots with the same features are used for transportation in each cell, with one robot in each cell. In addition, a robot with a different speed is used for intercellular transportation. If material needs to be transported in the same cell at the same time, there will be waiting times. In other words, only one part can be transported in a cell at one time. Similarly, parts that need to be transported to different cells cannot be transported at the same time. For a part to be transported to a different cell, the robot used in intercellular transportation must be available. Thus, in this study, the CF, CL, and scheduling problems are addressed in an integrated manner by considering the material handling system and alternative routes. A mathematical model and hyperheuristic (HH) algorithm are suggested for the problem. Because studies that consider CF, CL, and scheduling problems neglect the material handling system, the aim of this study is to find a solution to the integrated problem by also considering the material handling system and alternative routes. From an industrial perspective, the proposed framework is particularly relevant for CMSs such as automotive and electronics assembly systems, where robot-based material handling is being increasingly adopted. By explicitly modeling intracell and intercell transportation with heterogeneous robot speeds, the approach enables more accurate

scheduling, mitigates material handling congestion, and improves the utilization of automated transport resources.

The main contributions of this study are summarized as follows:

- A fully integrated decision framework is proposed that simultaneously addresses CF, CL, and scheduling while explicitly considering the material-handling system.
- A novel robot-based material-handling structure is introduced, in which identical robots operate within each cell, while a distinct robot with a different traveling speed is assigned for intercell transportation.
- Realistic transportation constraints are incorporated, including robot availability, non-overlapping transport operations, and waiting times when multiple parts request the same transportation resource. This results in a problem formulation that explicitly schedules robot movements.
- A novel mixed integer linear programming (MILP) model is proposed to optimally solve the addressed problem for small- and medium-sized instances.
- Alternative routing options are embedded within the integrated framework, enabling parts to follow different feasible routes.
- A hyperheuristic (HH) solution methodology is developed, together with a tailored fitness evaluation mechanism. The heuristic is designed independently of the mathematical model and was benchmarked not only against other heuristics but also against the MILP model on publicly available test problems.

The second section of the article presents a literature review, and the problem and mathematical model are presented in the third section. In the fourth section, the heuristic algorithm is provided, and the fifth section discusses the analysis of the proposed method. The last section concludes this work.

2. Literature review

2.1. Studies of integrated CF, CL, and scheduling problems

Few studies have considered the CF, CL, and scheduling problems together. Forghani and Ghomi [6] considered integrated CF, CL, and scheduling problems for virtual and classic CMS and considered alternative routes. The objective function of the study was to minimize the handling cost and cycle time. Computer software was designed for the problem, and the simulated annealing (SA) algorithm outperformed the other heuristics. Ebrahimi et al. [7] proposed a genetic algorithm (GA) for the integrated problem, and the group layout and scheduling problems were solved sequentially and concurrently. According to the results, improvements were made with the concurrent solution to the problem. The problem has a single objective function, and the authors considered alternative processing routes. Rahimi et al. [3] proposed a mixed integer programming (MIP) model for integrated problems to minimize the total completion time. They considered variable cell sizes, alternative processing routes, machine duplication, and reentrant parts. Fahmy [8] proposed a mixed integer linear programming (MILP) model for integrated CF, CL, and scheduling problems with sequence-dependent setups. The intercellular and intracellular transportation times were considered, and the objective function involved the minimization of the makespan or mean flow time. Arkat et al. [9] proposed two GA and MILP models to integrate the CF, CL, and scheduling problems while minimizing the total completion time. They reported that compared with addressing the problems separately, the use of the

integrated approach improved the performance of CMS. Arkat et al. [10] considered the integrated problem in a multi-objective manner. The objective functions involved the minimization of the total transportation cost and makespan, and a multi-objective heuristic algorithm was proposed to solve the problem. Wu et al. [1] proposed a mathematical model and a GA for integrating the CF, CL, and scheduling problems, and Motahari et al. [2] proposed a multi-objective algorithm in which the objective functions involved the minimization of the transportation cost, weighted completion time, and idle time. Heydari et al. [11] proposed a hybrid multi-objective algorithm to address the integrated CF, intracellular layout, and production planning problem, with the objectives of minimizing cost and production waste. Their solution approach combines branch-and-bound, genetic, and augmented ε-constraint methods to handle large-scale instances.

Although several integrated CF–CL–scheduling studies incorporate transportation or material handling aspects, they typically model them in an aggregated manner or as simple time/cost parameters rather than as explicit resource-constrained systems. For instance, Fahmy [8] considered intercellular and intracellular transportation times; however, the material-handling process was not modeled as a limited resource, and issues such as robot availability, waiting, or concurrent transport conflicts were not addressed. In other words, the scheduling of the material handling system was not explicitly considered. Similarly, Forghani and Ghomi [6], Arkat et al. [9,10], and Motahari et al. [2] included transportation times or costs and, in some cases, alternative routes; however, they did not explicitly represent the material-handling system as a robotic transportation process with capacity constraints. In contrast, this study introduces a robot-based material-handling structure that explicitly distinguishes between intracell and intercell transportation. It also assigns different travel speeds to these operations and incorporates availability, non-overlapping transport, and waiting constraints. To our knowledge, such a heterogeneous, resource-constrained robotic transportation model has not been previously integrated into CF–CL–scheduling formulations.

2.2. Studies of integrated CF and CL problem

In some studies, the CF and CL problems have been considered in an integrated manner. Bayram and Sahin [12] considered the CF and CL problem with alternative routes, equal machine dimensions, and multiple periods. They proposed an SA with linear programming and a GA to solve the problem. Forghani et al. [13] considered the CF and CL problem with respect to assembly and energy, and Chandrasekar and Venkumar [14] proposed an SA algorithm to solve the integrated CF and CL problem. Feng et al. [15] considered integrated CF and CL with alternative process routings and unequal machine dimensions. The SA algorithm was used to solve the problem. Forghani et al. [16] addressed the CF and CL problem with a multirow machine arrangement. They considered unequal machine dimensions, and an optimization model and a heuristic algorithm were proposed. Forghani et al. [17] addressed an SA algorithm for the integrated CF and CL problem, and Mahdavi et al. [18] proposed a flow matrix-based heuristic algorithm for the integrated CF and CL problem. Salimpour et al. [19] suggested a modified multi-objective algorithm for the CF and semi-robust CL problem in which the total material handling cost and dissimilarity of the machines in each cell were minimized. Jabal-Ameli et al. [20] considered the integrated CF and CL problem in a multi-objective manner with alternative routes and proposed a multi-objective scatter search algorithm.

2.3. Studies of integrated CF and scheduling problem

Some studies considered integrated CF and part scheduling problems. Alimian et al. [21] addressed the integrated CF and part scheduling problem with a preventive maintenance planning problem, and an optimization model was proposed. The cost of manufacturing was decreased considering the preventive maintenance planning problem. Ghezavati and Saidi-Mehrabad [22] proposed GA and SA algorithms for the problem with stochastic processing times, and Karthikeyan et al. [23] considered the CF problem. After obtaining a solution to the CF problem, the scheduling problem was considered, and the CF and scheduling problems were solved sequentially. Liu et al. [24] addressed the CF and scheduling problem while considering worker assignments. A discrete bacteria foraging algorithm was proposed to solve this problem. Papaioannou and Wilson [25] proposed fuzzy extensions to optimization models for CF and scheduling problems. In most of the studies, nonlinear mathematical models were proposed to solve the integrated CF and scheduling problem. Rafiei et al. [26] proposed a nonlinear mathematical model, SA, and GA to solve the integrated CF and scheduling problem, and Wang et al. [27] considered the integrated CF and scheduling problem and aimed to minimize the total tardiness penalty cost. A nonlinear mathematical model was proposed for the problem. Delgoshaei et al. [28] proposed a hybrid GA–SA algorithm for the machine location and part allocation problem while considering the drawbacks associated with cell load variation. A nonlinear mixed-integer programming model was formulated with the objective of minimizing the total cost.

In recent years, sustainability has increasingly been considered in CF problems [29]. Almasarwah et al. [30] proposed a stochastic mathematical model that addresses not only the manufacturing of final products but also the remanufacturing of returned products while allowing for machine duplication. In addition, a simulation model was developed to handle demand uncertainty.

3. Problem definition and mathematical model

3.1. Problem definition

In this study, an integrated CF, CL, and scheduling problem is considered. The problem is defined considering a world-class truck manufacturer. Material-handling robots are used to transport parts between machines. Parts may be transported between machines within the same cell or in different cells. In each cell, there is a robot to transport parts within the cell. Only one part can be transported by each material-handling robot at a time. The transportation of parts between different cells is conducted by a common robot, which is located in a corridor. Only one part can be transported between cells by this material-handling robot at a time. The speed of the robots in the cells is identical and is denoted by r, but the speed of the robot in the corridor is different than that of the other robots and is denoted by f. In the CF problem, parts and machines are assigned to cells by considering the similarities in production processes. In the CL problem, the arrangement of the cells within the shop floor and the optimal arrangement of machines in the cells are determined. In this study, a continuous CL problem is considered; in other words, it is assumed that the machines may have different sizes. Cx_m is the length, and Cy_m is the width of machine m. FL represents the floor length, and FW represents the floor width. In addition, each part has an alternative route, and a route should be determined for each part. $O_{p,r}$ denotes the number of operations of part p in route r. For example, if the operations of route r for part p are M1-M2-M3, the $O_{p,r}$ value of part p is 3. A part should be

processed on the machines sequentially, considering the determined route, to become a final product. $a_{p,r,s,m}$ is equal to 1 if the operation s of part p should be processed on machine m for route r; otherwise, it takes a value of 0. In this study, it is assumed that the cells are placed from bottom to top. In addition, the vertical coordinates of each cell are determined (ce_k), and machines are placed in a single-line layout structure in each cell. The common corridor on the left is used to handle parts between cells. The parts are processed on machines while considering the selected route. $P_{p,s,m}$ denotes the processing time of part p for operation s on machine m. Furthermore, the job shop scheduling environment is considered. In the job shop scheduling environment, each part has a different route to become a product. The material handling time between machines affects the completion time of parts, and the completion time of parts affects the scheduling problem. The material handling time is related to the CL and CF problems. Therefore, each problem is interrelated. The objective function involves the minimization of the maximum completion time (makespan). In addition, the machine gravity centers are accounted for when calculating the distance between machines, and the space between adjacent machines must be equal to or greater than the minimum required space between machines (G). Within each cell, there is an aisle for intracellular material flow through which parts can be transported by the robot. Material flow between machines in different cells is arranged by using the aisle on the left and the common robot. In this study, it is assumed that there is enough space in the cells and in the common corridor for the robots to move.

An example is provided in Figure 1. Similar layouts were also used by Vitayasak and Pongcharoen [4] and Feng et al. [15]. Machines M2–M1–M3–M4 are assigned to cell 1, machines M5–M8-M9 to cell 2, and machines M10–M6–M7–M11 to cell 3.

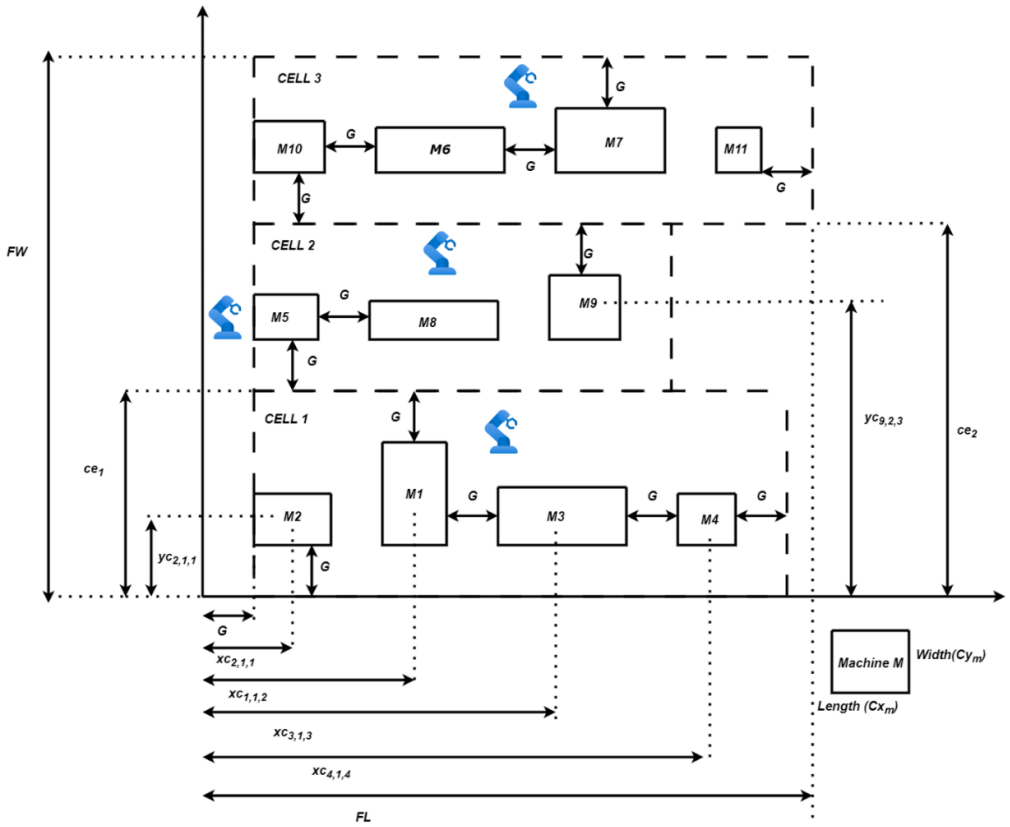


Figure 1. Layout type.

Rectilinear distance measurements between machines are used in this study. There are two cases when calculating the distance between machines.

1) Assume that machines m and m' are in the same cell k and that the sequences of machines in the cell are l and l', respectively. The calculation of the distance between machines m and m' $(d_{m,m'})$ is expressed in Equation 1.

$$d_{m,m'} = \quad |xc_{mkl} - xc_{m'kl'}| \tag{1}$$

2) If the m and m' machines are in cells k and k' and sequences l and l', respectively, then the distance between m and m' $(d_{m,m'})$ is calculated using Equation 2.

$$d_{m,m'} = \begin{cases} xc_{mkl} + xc_{m'kl'} + |ce_k - yc_{mkl}| + |yc_{m'k'l'} - ce_k|; \ If\ (k < k') \\ xc_{mkl} + xc_{m'kl'} + |ce_{k'} - yc_{mkl}| + |yc_{m'k'l'} - ce_{k'}|; \ If\ (k' < k) \end{cases} \tag{2}$$

3.2. Mathematical model

Sets and indices

$j', j,\ k',\ k$ *denote the cell indices,* $K = \{j', j, k', k | k = 1, .., N\}$

$m, g, g',\ m'$ *denote the machine indices,* $M = \{m, g, g'\ m' | m = 1, .., T\}$

$s, d, s',\ d'$ *denote the operation indices,* $S = \{s, d, s', d' | s = 1, .., G\}$

r', r *denote the route indices,* $R = \{r, r' | r' = 1, .., F\}$

o, p', p *denote the part indices,* $O = \{p, o, p' | o = 1, .., Z\}$

$l,\ l'$ *denote the machine sequence indices in cells,* $Q = \{l,\ l' | l = 1, .., T\}$

$t,\ t'$ *denote the position of the part indices on machines,* $S = \{t, t' | t = 1, .., Z\}$

Parameters

O_{pr}: $The\ number\ of\ operations\ of\ part\ p\ for\ route\ r$

a_{prsm}: $\begin{cases} 1; If\ operation\ s\ of\ part\ p\ of\ route\ r\ can\ be\ processed\ on\ machine\ m \\ 0; \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{cases}$

Cx_m: $The\ length\ of\ machine\ m$

Cy_m: $The\ width\ of\ machine\ m$

FL: Floor length

FW: Floor width

P_{prsm}: $Processing\ time\ of\ part\ p\ of\ route\ r\ for\ operation\ s\ and\ machine\ m$

G: Clearance between machines

r: The speed of the robot used for transporting material within a cell

f: The speed of the robot used for transporting material between cells

M: Large number

Decision variables

x_{prsmk}: $\begin{cases} 1; If\ operation\ s\ of\ part\ p\ in\ route\ r\ is\ processed\ on\ machine\ m\ and\ in\ cell\ k \\ 0; \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{cases}$

z_{psmt}: $\begin{cases} 1; If\ operation\ s\ of\ part\ p\ is\ processed\ on\ machine\ m\ on\ position\ t \\ 0; \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{cases}$

y_{mkl}: $\begin{cases} 1; If\ machine\ m\ is\ assigned\ to\ cell\ k\ on\ sequence\ l \\ 0; \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{cases}$

re_{pr}: $\begin{cases} 1; If\ route\ r\ is\ selected\ for\ part\ p \\ 0; \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{cases}$

xa_{psk}: $\begin{cases} 1; If\ transportation\ of\ part\ p\ for\ operation\ s\ is\ conducted\ by\ the\ robot\ in\ cell\ k \\ 0; \qquad Otherwise \end{cases}$

xb_{ps}: $\begin{cases} 1; If\ transportation\ of\ \ p\ for\ operation\ s\ is\ conducted\ by\ the\ common\ robot\ in\ the\ corridor \\ 0; \qquad Otherwise \end{cases}$

xc_{mkl}: $The\ coordinate\ of\ the\ x-axis\ of\ machine\ m\ assigned\ to\ k\ on\ sequence\ l$
yc_{mkl}: $The\ coordinate\ of\ the\ y-axis\ of\ machine\ m\ assigned\ to\ k\ on\ sequence\ l$
ce_k: $The\ coordinate\ of\ the\ y-axis\ of\ cell\ k$

$dx_{ps(s-1)mm'}$: Total intracell flow distance of part p between machine m and m' $for\ operation\ s$

$dy_{ps(s-1)mm'}$: Total intercell flow distance of part p between machine m and m' $for\ operation\ s$

CT_{psm}: $Completion\ time\ of\ part\ p\ for\ operation\ s\ on\ machine\ m$
ST_{psm}: $Processing\ starting\ time\ of\ part\ p\ for\ operation\ s\ on\ machine\ m$
C_p: $Completion\ time\ of\ part\ p$
TS_{ps}: $Transportation\ starting\ time\ of\ part\ p\ before\ operation\ s$
TC_{ps}: $Transportation\ completion\ time\ of\ part\ p\ before\ eration\ s$
C_{max}: $Makespan$

Model

$$Min\ Z = C_{max} \tag{3}$$

$$x_{prsmk} \le re_{pr} \times \sum_l a_{prsm} \times y_{mkl} \qquad \forall(k,s,r,m,p) \tag{4}$$

$$\sum_m \sum_k x_{prsmk} = re_{p,r} \qquad \forall(s \le O_{p,r}, r, p) \tag{5}$$

$$\sum_r re_{pr} = 1\ \forall(p) \tag{6}$$

$$\sum_k \sum_l y_{mkl} = 1 \quad \forall(m) \tag{7}$$

$$\sum_m y_{mkl} \le 1 \qquad \forall(k,l) \tag{8}$$

$$\sum_m y_{mkl} \le \sum_{m'} y_{m'k(l-1)} \qquad \forall(k, l>1) \tag{9}$$

$$xc_{mk1} \ge \frac{Cx_m}{2} + G - M \times (1 - y_{mk1}) \quad \forall(m,k) \tag{10}$$

$$xc_{mkl} \ge xc_{m'k(l-1)} + \frac{Cx_{m'}}{2} + \frac{Cx_m}{2} + G - M \times (2 - y_{mkl} - y_{m'k(l-1)})$$

$$\forall(m,m',k,l>1, m \ne m') \tag{11}$$

$$yc_{m1l} \ge \frac{Cy_m}{2} + G - M \times (1 - y_{m1l}) \quad \forall(m,l) \tag{12}$$

$$yc_{m1l} \le \frac{Cy_m}{2} + G + M \times (1 - y_{m1l}) \quad \forall(m,l) \tag{13}$$

$$yc_{mkl} \ge ce_{k-1} + \frac{Cy_m}{2} + G - M \times (1 - y_{mkl}) \quad \forall(m,l,k>1) \tag{14}$$

$$ce_1 \geq Cy_m + 2 \times G - M \times (1 - y_{m,1,l}) \quad \forall(m, l) \tag{15}$$

$$ce_k \geq ce_{k-1} + Cy_m + 2 \times G - M \times (1 - y_{mkl}) \quad \forall(m, l, k > 1) \tag{16}$$

$$dx_{ps(s-1)mm'} \geq |xc_{mkl} - xc_{m'kl'}| - M \times (4 - y_{mkl} - y_{m'kl'} - x_{pr(s-1)m'k} - x_{prsmk})$$

$$\forall(k, l, m', s, r, p, m, l' \text{ and } m \neq m', l' < l,) \tag{17}$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} + |ce_k - yc_{mkl}| + |yc_{m'k'l'} - ce_k| -$$

$$M \times (2 - x_{pr(s-1)m'k'} - x_{prsmk}) \quad \forall(p, m', s, r, m, l, l' \text{ and } k < k') \tag{18}$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} + |ce_{k'} - yc_{mkl}| + |yc_{m'k'l'} - ce_{k'}| -$$

$$M \times (2 - x_{prsmk} - x_{pr(s-1)m'k'}) \quad \forall(p, r, m, m', l, l', k, k', s \text{ and } k' < k) \tag{19}$$

$$\sum_t z_{psmt} \leq 1 + M \times (1 - x_{prsmk}) \quad \forall(p, r, s, m, k) \tag{20}$$

$$\sum_t z_{psmt} \geq 1 - M \times (1 - x_{prsmk}) \quad \forall(p, r, s, m, k) \tag{21}$$

$$\sum_t \sum_m z_{psmt} \leq 1 \quad \forall(p, s) \tag{22}$$

$$\sum_p \sum_s z_{psmt} \leq 1 \ \forall(m, t) \tag{23}$$

$$\sum_p \sum_s z_{psmt-1} \geq \sum_{p'} \sum_{s'} z_{p's'mt} \quad \forall(m, t > 1) \tag{24}$$

$$CT_{psm} \geq ST_{psm} + P_{prsm} - M \times (2 - z_{psmt} - re_{pr}) \forall(t, m, s, r, p) \tag{25}$$

$$ST_{psm} \geq CT_{p's'm} - M \times (2 - z_{psmt} - z_{p's'm(t-1)}) \ \forall(p, p', s > 1, s', m, t) \tag{26}$$

$$ST_{psm} \geq CT_{p(s-1)m'} - M \times (2 - z_{psmt} - z_{p(s-1)m't'}) \quad \forall(t, s > 1, m, m', p, t') \tag{27}$$

$$C_p \geq CT_{psm} \quad \forall(p, s, m) \tag{28}$$

$$TC_{ps} \geq TS_{ps} + \left(\frac{dx_{ps(s-1)mm'}}{r} + \frac{dy_{ps(s-1)mm'}}{f}\right) - M \times (2 - z_{psmt} - z_{p(s-1)m't'})$$

$$\forall(m', p, t', s > 1, m, t) \tag{29}$$

$$TS_{ps} \geq CT_{p(s-1)m} \quad \forall(p, s > 1, m) \tag{30}$$

$$ST_{psm} \geq TC_{ps} \quad \forall(p, s, m) \tag{31}$$

$$TS_{p's'} \geq TC_{ps} - M \times (2 - xa_{psk} - xa_{p's'k}) - M \times \left(1 - fi_{pp'ss'}\right)$$

$$\forall(s' > 1\ , p, p', s > 1\ and\ k) \tag{32}$$

$$TS_{ps} \geq TC_{p's'} - M \times fi_{pp'ss'} - M \times (2 - xa_{p's'k} - xa_{psk})$$

$$\forall(p, s > 1, p', s' > 1, k) \tag{33}$$

$$TS_{p's'} \geq TC_{ps} - M \times \left(1 - fo_{pp'ss'}\right) - M \times (2 - xb_{ps} - xb_{p's'})$$

$$\forall(p, s > 1, s' > 1, p') \tag{34}$$

$$TS_{ps} \geq TC_{p's'} - M \times fo_{pp'ss'} - M \times (2 - xb_{ps} - xb_{p's'})$$

$$\forall(s > 1, p', p, and\ s' > 1) \tag{35}$$

$$\sum_t \sum_m \sum_s z_{psmt} \leq M \times \left(1 - re_{pr}\right) + Op_{pr}\ \forall(r, p) \tag{36}$$

$$xc_{mkl} + \frac{Cx_m}{2} + G \leq FL \qquad \forall(l, k, m) \tag{37}$$

$$yc_{mkl} + \frac{Cy_m}{2} + G \leq FW \qquad \forall(m, k, l) \tag{38}$$

$$xa_{psk} \leq 1 + M \times (2 - x_{prsmk} - x_{pr(s-1)m'k})\ \ \forall(m, p, s > 1, k, r, m') \tag{39}$$

$$xa_{psk} \geq 1 - M \times (2 - x_{prsmk} - x_{pr(s-1)m'k})\ \ \forall(p, s > 1, k, r, m, m') \tag{40}$$

$$xb_{ps} \leq 1 + M \times (2 - x_{prsmk} - x_{pr(s-1)m'k'})\ \ \forall(p, r, m, m', k, k's > 1,)\ k \neq k' \tag{41}$$

$$xb_{ps} \geq 1 - M \times (2 - x_{pr(s-1)m'k'} - x_{prsmk})\ \ \forall(k, r, k', m, p, s > 1, m')k \neq k' \tag{42}$$

$$M \times y_{mkl} \geq xc_{mkl} \qquad \forall(l, m, k) \tag{43}$$

$$M \times y_{mkl} \geq yc_{mkl} \qquad \forall(l, m, k) \tag{44}$$

$$C_{max} \geq C_p \quad \forall(p) \tag{45}$$

$$x_{prsmk},\ y_{mkl},\ re_{pr}, xb_{ps}, xa_{psk} \in \{0,1\} \tag{46}$$

$$xc_{mkl},\ yc_{mkl},\ ce_k,\ dx_{pss'mm'},\ dy_{pss'mm'}, C_{max} \geq 0 \tag{47}$$

Equation 3 is the objective function and minimizes the makespan, and Equation 4 ensures that the operations of the parts are assigned to cells while considering the route and machines. Equation 5

ensures that the operation of a part is assigned to a cell and machine while considering the selected route. Equation 6 ensures that a route is selected for each part, and Equation 7 ensures that each machine is assigned to a sequence of a cell. Equation 8 ensures that at most one machine may be assigned to each sequence of cells, while Equation 9 ensures that the machines are assigned to the cells sequentially. Equation 10 is used to calculate the coordinates of the x-axis of the machines assigned to the first sequence of each cell. Equation 11 is used to calculate the coordinates of the x-axis of machines assigned to sequences other than the first sequence of a cell. Equations 12 and 13 are used to calculate the coordinates of the y-axis of the machines in the first cell, Equation 14 is used to calculate the coordinate of the y-axis of machines assigned to a cell other than the first, Equation 15 is used to calculate the coordinate of the y-axis of the cell in the first row, and Equation 16 is used to calculate the coordinates of the y-axis of the other cells. Equation 17 is used to calculate the flow distance between machines in the same cell, while Equations 18–19 are used to calculate the flow distance between machines in different cells. Equations 20 and 21 ensure that if a part is assigned to a machine in a cell, it is assigned to a position in the machine, and Equation 22 ensures that the operation of a part is assigned to a maximum of one machine. Equation 23 ensures that at most one part is assigned to a position of each machine. Equation 24 ensures the sequential assignment of jobs to machine positions. Equation 25 is used to calculate the completion time of parts for operations. Equations 26–27 are used to calculate the processing starting time of the parts, and Equation 28 is used to calculate the completion time of the parts. Equation 29 is used to calculate the transportation completion time of parts. Equation 30 ensures that after an operation on a part is completed, transportation for the next operation may start. Equation 31 ensures that the part is transported to a machine before it can be processed on that machine. Equations 32–33 ensure the transportation of only one part at a time by using the robots in each cell, and Equations 34–35 ensure the transportation of only one part at a time with the common robot. Equation 36 ensures that parts are assigned to the number of machines considering the number of operations. Equation 37 establishes that the coordinates of the x-axis of the cells are less than FL, and Equation 38 establishes that the coordinate of the y-axis of the cells is less than FW. Equations 39 and 40 ensure that the robot in the relevant cell transports the part if the consecutive operations of the part are to be performed in the same cell, and Equations 41 and 42 ensure that the common robot transports the part if the consecutive operations of the part are to be performed in different cells. Equations 43 and 44 ensure that the x-coordinate and y-axis in the relevant cell are calculated if the machine is assigned to that cell. Finally, Equation 45 is used to calculate the objective function of the problem, and Equations 46 and 47 are sign constraints.

3.3. Linearization of the model

The model becomes nonlinear due to Equations 4, 17, 18, and 19. A new, auxiliary binary variable Le_{prmk} is defined for linearization. Additionally, Equations 48–50 are added to replace Equation 2.

$$x_{prsmk} \leq Le_{prmk} \times a_{prsm} \qquad \forall(p,r,s,m,k) \tag{48}$$

$$2 \times Le_{prmk} \leq re_{pr} + \sum_l y_{mkl} \, \forall(r,k,m,p) \tag{49}$$

$$Le_{prmk} + 1 \geq re_{pr} + \sum_l y_{mkl} \, \forall(r,k,m,p) \tag{50}$$

Finally, Equations 51–56 are added to replace Equations 17, 18, and 19, and the model was made linear.

$$dx_{ps(s-1)mm'} \geq xc_{mkl} - xc_{m'kl'} - M \times (4 - x_{prsmk} - x_{pr(s-1)m'k} - y_{m'kl'} - y_{mkl})$$
$$\forall (k, r, m', l, p, s, m, l' \; and \; l' < l, m \neq m') \quad (51)$$

$$dx_{ps(s-1)mm'} \geq -xc_{mkl} + xc_{m'kl'} - M \times (4 - x_{pr(s-1)m'k} - x_{prsmk} - y_{m'kl'} - y_{mkl})$$
$$\forall (k, r, m', l, p, s, m, l' \; and \; m \neq m', l < l') \quad (52)$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} - M \times (2 - x_{pr(s-1)m'k'} - x_{prsmk}) + (ce_k - yc_{mkl}) +$$
$$(yc_{m'k'l'} - ce_k) \qquad \forall (p, m', s, m, l, l', r \; and \; k < k') \quad (53)$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} - M \times (2 - x_{pr(s-1)m'k'} - x_{prsmk}) + (-ce_k + yc_{mkl}) +$$
$$(-yc_{m'k'l'} + ce_k) \quad \forall (p, m', s, m, l, l', r \; and \; k < k') \quad (54)$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} - M \times (2 - x_{pr(s-1)m'k'} - x_{prsmk}) + (ce_{k'} - yc_{mkl}) +$$
$$(yc_{m'k'l'} - ce_{k'}) \qquad \forall (p, m', s, m, l, l', r \; and \; k < k') \quad (55)$$

$$dy_{ps(s-1)mm'} \geq xc_{mkl} + xc_{m'kl'} - M \times (2 - x_{p,r,(s-1),m',k'} - x_{p,r,s,m,k}) + (-ce_{k'} + yc_{mkl}) +$$
$$(-yc_{m'k'l'} + ce_{k'}) \quad \forall (k', p, m', k, s, m, l, l', r \; and \; k' < k) \quad (56)$$

3.4. Example problem

There are 4 parts, 2 routes for each part, 4 machines, and 2 cells. The O_{pr}, a_{prsm}, and P_{psm} parameters are provided in Table 1, and the processing times, given with ( ), can also be found in Table 1. The (Cx, Cy) values for each machine are (2,3), (4,2), (5,4), and (3,5). FL and FW are 12, and the speed of the robots is 2 units. The clearance between machines is 1.

Table 1. O_{pr}, a_{prsm}, and P_{prsm} parameters.

Parts	Route 1	Route 2
1	M1(2)-M2(1)-M4(4)	M2(1)-M3(3)-M4(4)
2	M2(1)- M4(4)	M1(2)-M2(1)-M3(3)
3	M3(3)-M1(2)	M1(1)- M4(4)
4	M2(1)-M3(3)	M1(1)-M3(3)

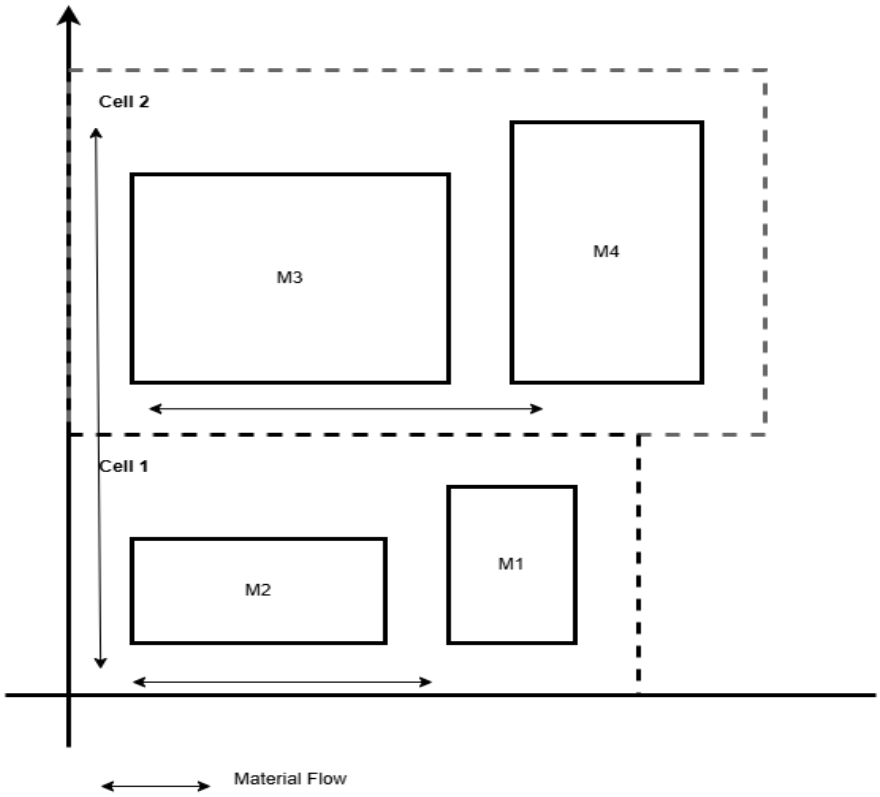


Figure 2. Layout of the obtained solution.

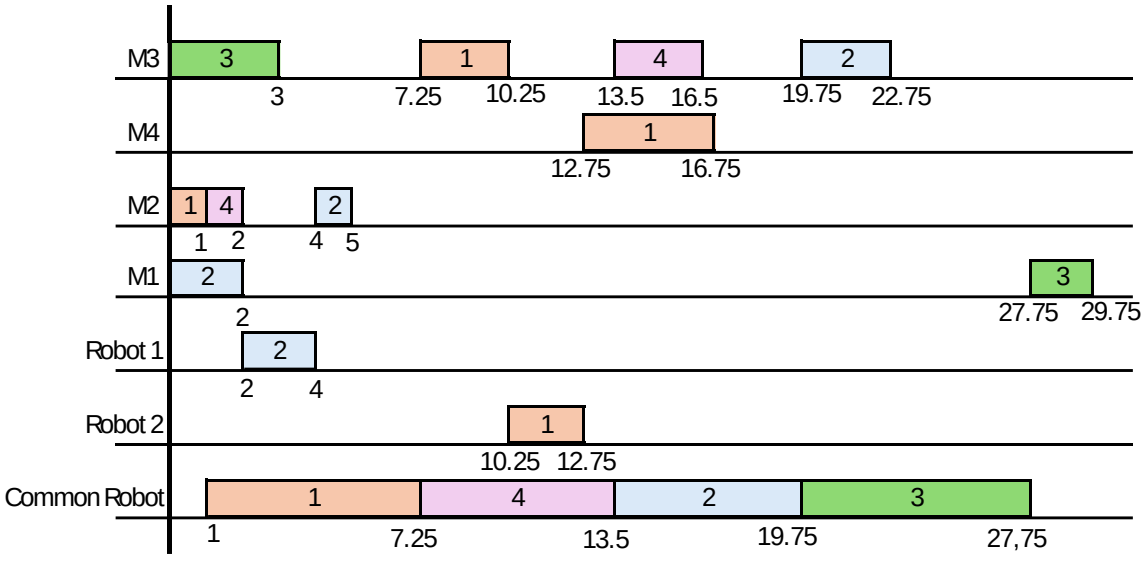


Figure 3. Processing and transportation sequences of parts on machines and robots.

Route 2 is selected for parts 1 and 2, and Route 1 is selected for parts 3 and 4. The total flow distance between machines 2 and 3 is 12.5, that between machines 3 and 4 is 5, that between machines 1 and 2 is 4, and that between machines 3 and 1 is 16. The makespan is 29.75. The layout of the solution

is depicted in Figure 2, and the processing sequence of parts on related machines and the transportation sequence of parts on the robots are given in Figure 3.

In this study, the MILP formulation is introduced solely as an exact reference model to represent the problem in an optimal and rigorous manner. However, owing to the NP-hard nature of the problem, this formulation can be solved only for instances of limited size. For this reason, the heuristic-based solution approach is proposed to handle larger problem instances. Notably, the proposed heuristic is not designed to operate in cooperation with the MILP model. Instead, feasibility with respect to the problem structure is ensured through a dedicated decoding procedure embedded within the heuristic framework. By means of this decoding mechanism, valid solutions can be generated directly without relying on the MILP formulation. The results obtained from the MILP model are used only as a benchmark for small-scale instances to evaluate the performance of the heuristic approach.

4. Hyperheuristic algorithm

HH algorithms are high-level search methodologies that focus on problem-solving methods [31]. There are two classes of HH algorithms, as they either select existing heuristics or generate new heuristics. While heuristic-generating algorithms are generally designed on the basis of genetic programming, algorithms that select from a series of heuristics are designed on the basis of metaheuristic algorithms. HH aims to obtain solutions by effectively using many different heuristic algorithms. Many factors, such as the order of the heuristic algorithms, the parameters of the heuristics, and the neighborhood structures, affect the success of the heuristic algorithm [32]. Furthermore, these algorithms automatically make decisions, such as the order, parameter values, and neighborhood structures in which a series of heuristic algorithms will be used. HH algorithms also consist of two levels: high-level strategies and low-level strategies. The low-level strategy involves a series of heuristic algorithms, and the high-level strategy determines the order, parameters, and neighborhood structures in which the low-level algorithms will be used. While the high-level heuristic works in heuristic space, the low-level strategy works in the solution space. In this study, the low-level heuristic algorithms include the GA, artificial bee colony (ABC), and hill-climbing (HC) algorithms.

In the first subsection, the steps of the GA, ABC, and HC algorithms are discussed. In the second subsection, the HH approach is explained, and in the third section, the solution representation and decoding algorithm are provided. The proposed algorithm differs from the literature, which considered the decoding algorithm and HH approaches based on the GA, ABC, and HC algorithms.

The flowchart of the HH algorithm is given in Figure 4.

Low-level heuristics (GA, ABC, and HC) were selected because they rely on fundamentally different solution selection mechanisms, local optimum escape strategies, and intensification behaviors, which are essential for assignment and scheduling problems. The GA promotes diversification mainly through mutation, while promising individuals are selected using tournament selection. In contrast, ABC generates new candidate solutions via a population-based mechanism involving employed, onlooker, and scout bees, enabling adaptive exploration without rigid construction rules. Diversification in ABC is primarily achieved through the scout bee phase, where entirely new solutions are generated. HC complements these approaches with a purely greedy strategy that rapidly improves solutions through local moves. The combined use of the GA, ABC, and HC allows for the hyperheuristic framework to exploit complementary solution selection, diversification, and intensification strategies.

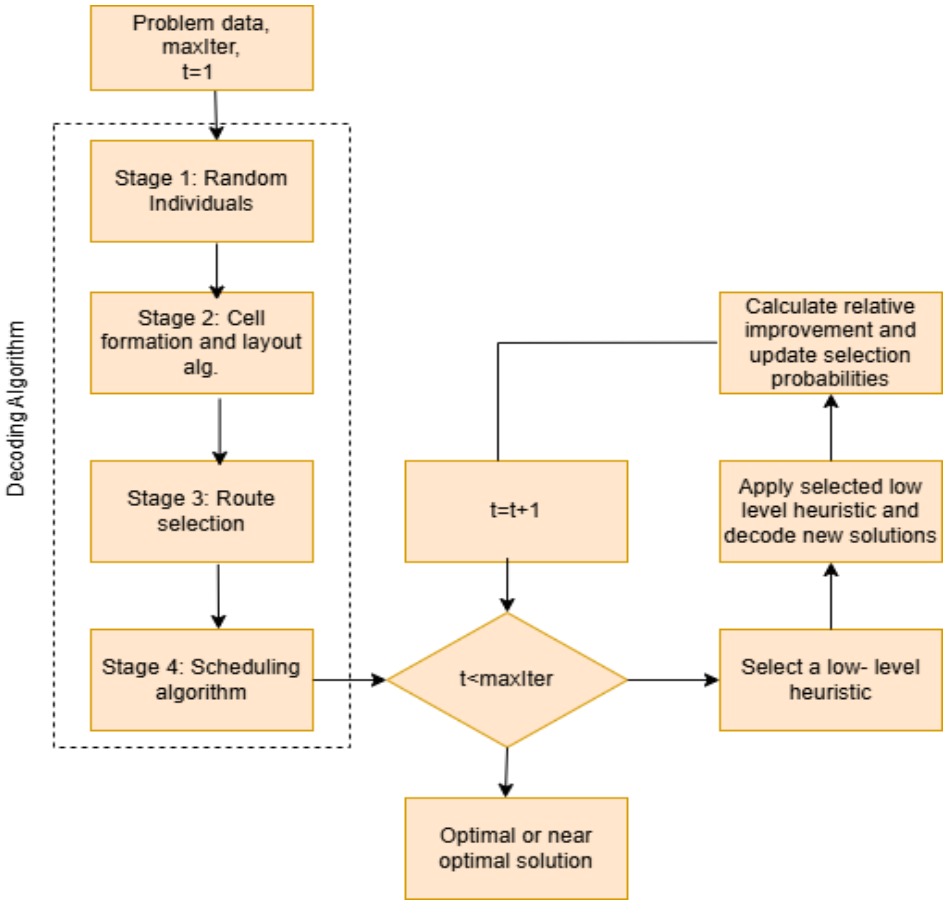


Figure 4. Flowchart of the proposed HH.

4.1. Low-level heuristics

4.1.1. Genetic algorithm

In the GA, the initial population with N_{pop} individuals is randomly generated, and the fitness value of each solution is calculated with the proposed decoding algorithm. The mating pool is obtained via tournament selection, and the solutions are randomly matched. Crossover and mutation operators are implemented in the population by considering the crossover rate (Cr) and mutation rate (Mu). These operators are also applied to the solutions in the permutation representation. A two-point crossover operator is used in the study. In the mutation operator, two randomly selected parts (machines) are replaced, and the fitness of the new population is calculated with the decoding algorithm. The algorithm is run until the number of generations (NP) is reached. Genetic algorithms do not rely on explicit acceptance rules; instead, solution survival is governed by fitness-driven selection mechanisms, where high-quality individuals are more likely to be retained in the population.

4.1.2. Artificial bee colony algorithm

The ABC algorithm is a swarm-based heuristic algorithm that was proposed by Karaboga [33], which simulates the foraging behavior of bees. The ABC algorithm has three stages: the employed bee

phase, the onlooker bee phase, and the scout bee phase [34]. First, N_{pop} solutions are generated randomly, and the fitness values are calculated. The $trial$ value of each solution is set to zero. In the employed bee phase, each solution is randomly matched with another solution, a two-point crossover operator is applied, and a new solution is obtained. If the new solution is better than the original solution, the original solution is replaced with the new solution, and the $trial$ value of this solution is set to zero. Otherwise, the $trial$ value is increased by 1. The fitness values of the solutions are calculated by Equation 57.

$$f(i) = \begin{cases} \frac{1}{1+Z(i)} & If\ Z(i) \geq 0 \\ 1 + |Z(i)| & If\ Z(i) < 0 \end{cases} \tag{57}$$

The probability value is calculated by Equation 58, where F is the maximum fitness value.

$$Probability(i) = 0{,}9 \times \frac{f(i)}{F} + 0{,}1 \tag{58}$$

In the onlooker bee stage, solutions are selected considering the probability values of the solutions. Each solution is randomly matched, a new solution is obtained by using two-point crossover, and the $trial$ values are calculated. In the scout bee phase, solutions with trial values greater than the $limit$ value are replaced with a random solution, and the iteration count is increased by 1. The algorithm is run until the iteration count is equal to $maxtrial$.

In both the ABC and GA procedures, offspring are generated through a two-point crossover mechanism applied to permutation-based encoding. The working principle of this operator is illustrated in Figure 5. Two cut positions are randomly selected along the machine and part sequences. For each child, the segments located before the first cut and after the second cut are directly inherited from one parent, while the remaining section is completed by preserving the relative order of the elements taken from the other parent. In this way, the feasibility of the permutation structure is maintained when information from both parents is combined.

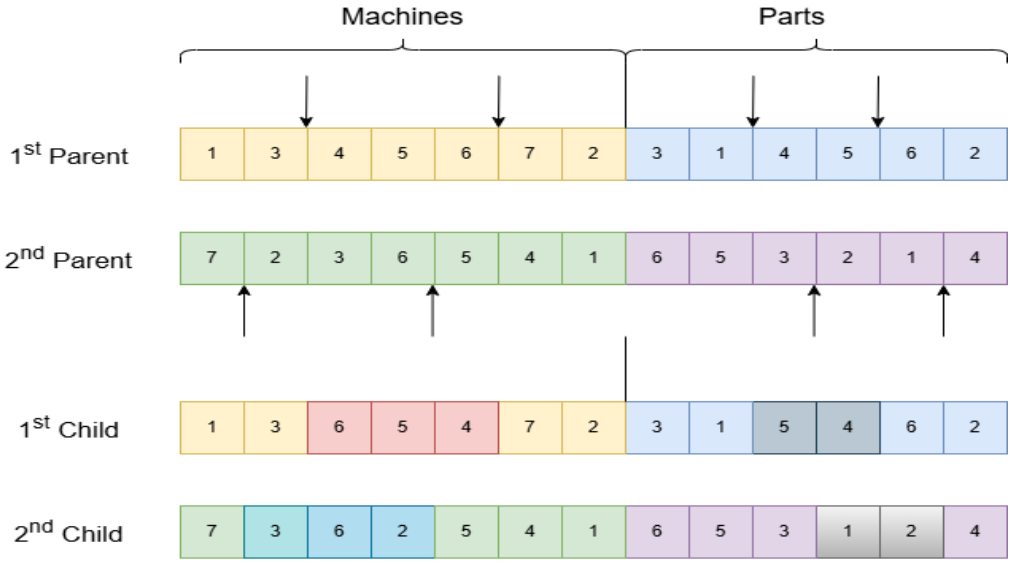


Figure 5. Two-point crossover.

4.1.3. Parallel hill-climbing algorithm

The HC algorithm is a local search algorithm that is used in many optimization problems [35,36]. In this study, the steepest mutation-based parallel HC algorithm is employed [37]. In this algorithm, N_{pop} solutions are generated, and new solutions are generated from the initial solutions with the swap operator. In the swap operator, a random value n is generated for each solution separately for the machine and part segments. Accordingly, n randomly selected positions are exchanged within the machine and part sections to generate new candidate solutions. The value of n is randomly drawn from the intervals $[\frac{T}{4}, \frac{T}{3}]$ for the machine section and $[\frac{Z}{4}, \frac{Z}{3}]$ for the part section, where T and Z denote the sizes of the machine and part sequences, respectively. If the new solutions are better than the initial solutions, the initial solutions are replaced by the new solutions, and the iteration count is increased by 1. The algorithm runs until the iteration count is equal to NP.

4.2. Hyperheuristic approach

In this study, three low-level heuristic algorithms, the ABC, GA, and HC, were chosen. With the use of high-level heuristics, the probability of choosing more successful algorithms is high [31]. Thus, in this study, a HH algorithm was designed with the method introduced by Wang et al. [37].

The relative improvements of each algorithm at iteration t are calculated by using Equation 59. $\eta_k(t)$ denotes the relative improvement of the k[th] algorithm for iteration t.

$$\eta_k(t) = \max\left\{0, \frac{Z_{best} - Z_{new}}{Z_{best}}\right\} \tag{59}$$

The selection probability of each algorithm is updated considering Equations 60–62. Δ_{HH} is the randomization ratio.

$$p_{GA}^{new}(t) = p_{GA}(t) + \Delta_{HH} + \eta_{GA}(t) \tag{60}$$

$$p_{ABC}^{new}(t) = p_{ABC}(t) + \Delta_{HH} + \eta_{ABC}(t) \tag{61}$$

$$p_{HC}^{new}(t) = p_{HC}(t) + \Delta_{HH} + \eta_{HC}(t) \tag{62}$$

Feedback mechanisms are implemented by using Equations 63–65, which are given below:

$$p_{GA}(t+1) = \frac{p_{GA}^{new}(t)}{p_{GA}^{new}(t) + p_{ABC}^{new}(t) + p_{HC}^{new}(t)} \tag{63}$$

$$p_{ABC}(t+1) = \frac{p_{ABC}^{new}(t)}{p_{GA}^{new}(t) + p_{ABC}^{new}(t) + p_{HC}^{new}(t)} \tag{64}$$

$$p_{HC}(t+1) = \frac{p_{HC}^{new}(t)}{p_{GA}^{new}(t) + p_{ABC}^{new}(t) + p_{HC}^{new}(t)} \tag{65}$$

$p_k(t+1)$ denotes the selection probability of the k[th] algorithm for iteration $t+1$.

The HH method is described below.

Procedure: Hyperheuristic

Input: Problem parameters and parameters of the algorithms

Output: Optimal or near-optimal solution
Set initial values of p_{GA}, p_{ABC}, and p_{HC} with equal probabilities
for i=1:maxIter
rand←random();
if (rand≤p_{GA})
select the GA;
elseif (rand>p_{GA})and (rand≤($p_{GA}+p_{ABC}$))
select the ABC;
else
select the HC;
end
Calculate the relative improvement;
Update the selection probabilities;
end

4.3. Solution representation and decoding algorithm

In the addressed problem, route assignment to each part, machine assignment to the cells, the determination of the appropriate layout, and job scheduling on the machines are conducted in an integrated approach while considering the material-handling robot. All of these decision problems should be solved to calculate the objective function value of a solution. A decoding algorithm with four stages is proposed to calculate the objective function of a random solution. In the first stage, the individuals in the population are randomly generated. In the second stage, the layout problem is considered, and in the third stage, the route for each job is assigned. In the fourth stage, the parts are scheduled on the machines considering the material handling system.

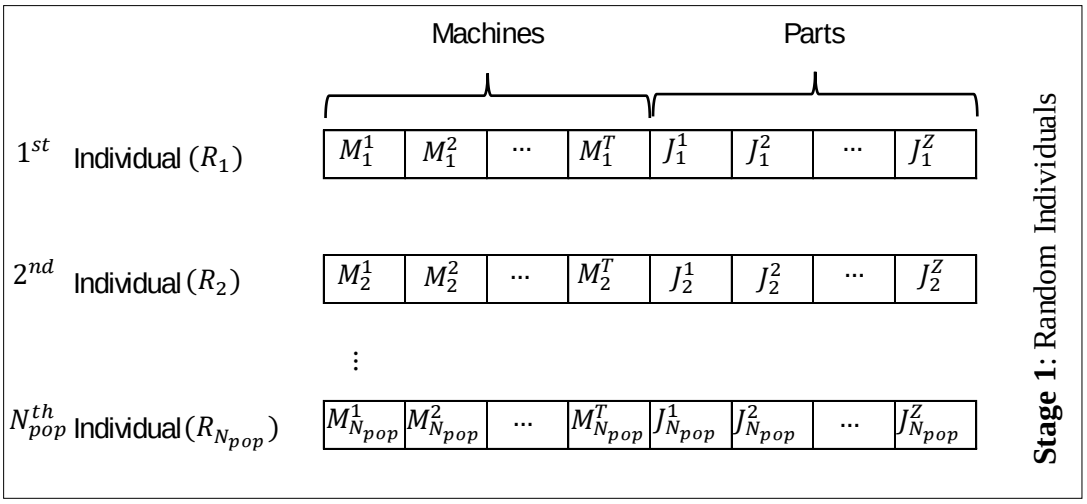


Figure 6. Stage 1: Random individuals.

4.3.1. Stage 1: Random individuals

The initial solution matrix of size (N_{pop}×(T+Z)) is randomly generated, with the population size (N_{pop}), the number of machines T, and the number of parts Z. In the matrix, the machines are randomly ordered in the first part, and the jobs are randomly ordered in the second part. The first T cells make up the first part, and the other sequential Z numbers of the cells constitute the second part. The generated matrix is denoted by R_i. A representation of the individuals is provided in Figure 6.

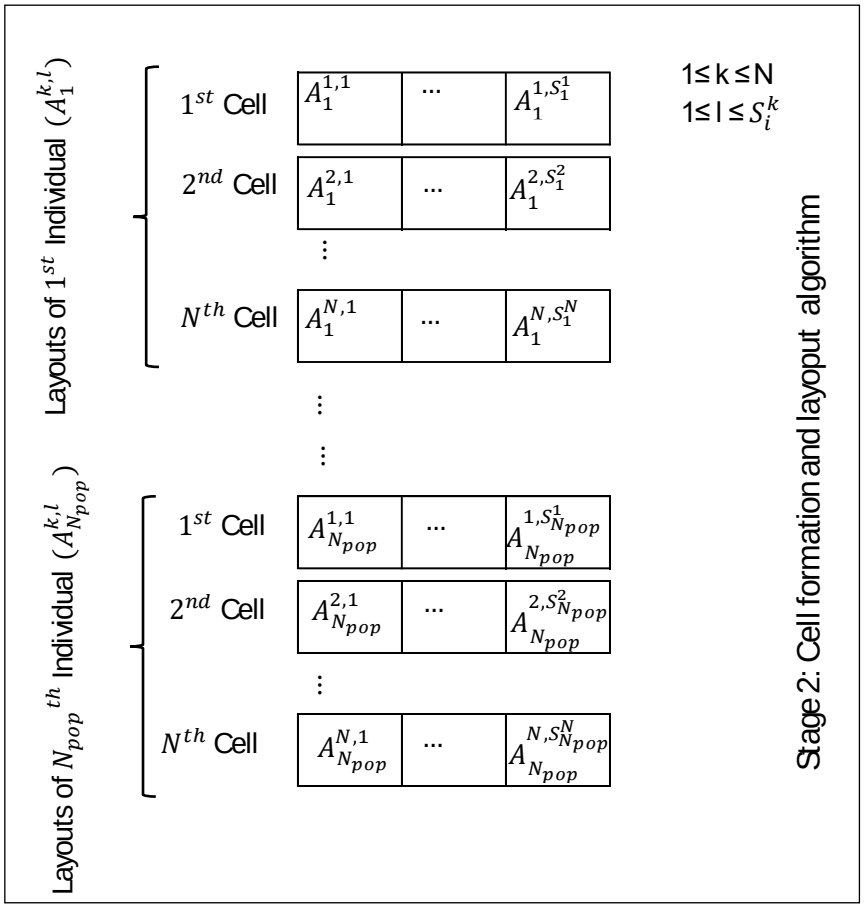


Figure. 7. Stage 2: Cell formation and layout algorithm.

4.3.2. Stage 2: Cell formation and layout algorithm

In the CF and layout algorithm, the random solutions generated in Stage 1 are considered. By using this algorithm, the machines are assigned to the cells, and the layout is determined. The section in which the machines are randomly ranked in the R_i matrix is considered. Starting from the first individual, the first machine is assigned to the randomly selected k^* cell, and $S_i^{k^*}$ is increased by 1. S_i^k denotes the number of machines assigned to cell k for individual i. After the assignment, the area of cell W_{k^*} that is being used is calculated, and the next machine is assigned to the cell k^* so that

the area used in the relevant cell does not exceed the FL value. Equation 66 is used to determine the cell k^* to which the next machine M should be assigned.

$$k^* \leftarrow \arg\max_{1\le k\le N}\{(FL - W_k)|W_k + G + Cx_M \le FL\} \tag{66}$$

If no cell that does not exceed the FL value for the related machine M is available, an infeasible solution is obtained, and the inf_i value of the relevant solution is set to 1. After all of the machines are assigned, if the inf_i value of the current solution is 0, the ce_k value of each cell is calculated. If the ce_N value of the last cell is less than FW, the relevant solution is a feasible solution and is recorded. Otherwise, the inf_i value is set to 1. After running the algorithm for all individuals, instead of individuals whose inf_i value is 1, the randomly selected feasible individuals whose inf_i value is 0 are recorded. Thus, a population with N_{pop} feasible solutions is obtained. An example representation after applying the algorithm is given in Figure 7.

Procedure: Cell formation and layout algorithm
Input: Random individuals (R_i), FL, FW, Cx_m, Cy_m, G, T, N_{pop}
Output: Assignment of machines to cells and layouts of cells $(A_i^{k,l})$
For i=1:N_{pop}
  $S_i^k \leftarrow 0,\ A_i^{k,l} \leftarrow 0,\ inf_i \leftarrow 0;\ W_k \leftarrow 0;$ $(1 \le k \le N\ and\ 1 \le l \le T)$
  For j=1:T
    $M \leftarrow M_i^j$;
    //Select the cell k^* with maximum empty space and the
    //assignment of machine M to cell k^* does not exceed the FL
    $k^* \leftarrow \arg\max_{1\le k\le N}\{(FL - W_k)|W_k + G + Cx_M \le FL\}; S_i^{k^*} \leftarrow S_i^{k^*} + 1;\ W_{k^*} \leftarrow W_{k^*} + G + Cx_M;$
    Assign machine M to cell $k^*$to $S_i^{k^*}$ position and $l \leftarrow S_i^{k^*};\ A_i^{k^*,l} \leftarrow M;$
    If$(k^* == \{\})$
      Infeasible solution and $inf_i \leftarrow 1;$
    End
  End
  If $(inf_i == 0)$
    For k=1:N
      If $(k == 1)$
        $ce_k \leftarrow \max_{1\le l\le S_i^k}\left(2G + Cy_{A_i^{k,l}}\right);$
      Else
        $ce_k \leftarrow ce_{k-1} + \max_{1\le l\le S_i^k}\left(2G + Cy_{A_i^{k,l}}\right);$
      End
      If $(ce_N > FW)$
        $inf_i \leftarrow 1;$
      End
    End
  End
  If $(inf_i$==0)
    Save $A_i^{k,l}$;
  End

End
For i=1: N_{pop}
If $(inf_i == 1)$
Select a random solution i^* with inf_{i^*}=0 and replace the layout of individual i with the i^* solution; $A_i^{k,l} \leftarrow A_{i^*}^{k,l}$:
End
End

4.3.3. Stage 3: Route selection

In this stage, a route is determined for each part. The distance between the machines in each route is calculated, and the route of each part is selected with the minimum total machine distance. In total, two solutions are created from each individual. One is the solution in which the routes are determined randomly, and the other is the solution in which the routes have the minimum machine distance. As a result, $2\times N_{pop}$ individuals are generated in this stage. The route of individual i for part J is denoted by $route_i^J$.

4.3.4. Stage 4: Scheduling algorithm considering material-handling robots

The scheduling of the parts is carried out by considering the selected route and layout of the cells. In the scheduling phase, the completion time and objective function of the parts are calculated so that each robot in the cells and in the common corridor can carry a single material at a time.

The objective function of the entire population is calculated starting from the first individual in the population. To calculate the objective function of individual i, first, the part in the R_i matrix for which the jobs are ranked and for the first job in order (T+1) is accounted for, and this part is assigned the value of J. N_J denotes the number of operations of part J and is assumed to be 1. Part J is processed on machine m^* for operation N_J, and the completion time of operation N_J for part J on machine m^* is calculated with Equation 67, which is provided below. P_{J,N_J,m^*} is the processing time of the operation N_J for part J and machine m^*. The first operations of all of the parts listed in the R_i matrix are assigned to the relevant machines, and CT_{m^*} is calculated. The completion time of operation N_J for part J $(PT_J^{(N_J)})$ is equal to CT_{m^*}.

$$CT_{m^*} \leftarrow CT_{m^*} + P_{J,N_J,m^*} \tag{67}$$

The operation count N_J of part J is increased by 1 unit $(N_J \leftarrow N_J + 1)$. Then, starting from the first part in R_i, the robot to be used for transportation for the N_J operation for all parts is determined. First, the transportation completion time of all robots TC_o is taken as zero. TC_o is the time at which the robot completes the transportation operation. The o index takes values between 1 and k+1. If o takes the value k+1, it means that the common robot will be used for transportation. If the o index takes the value k, the robot in cell k will be used for transportation. If the machine m^* used for the operation $(N_J - 1)$ and the machine m^{**} to be used in the operation N_J are in cell k^* for job J, the $TL_{k^*}^{n_{k^*}}$ value is calculated. $TL_{k^*}^{n_{k^*}}$ shows the completion time of the transportation operation of the alternative n_{k^*} in cell k^*. n_{k^*} is set to 1. $TL_{k^*}^{n_{k^*}}$ is calculated with Equation 68 below. While $TL_{k^*}^{n_{k^*}}$ is being calculated, the completion time of the robot's maximum value from the previous transportation

task and the completion time of the previous operation of the part are added to the distance between the two machines divided by r (the speed of the robot). Thus, the transportation completion time for alternative n_{k^*} on robot k^* is found. $As_{k^*}^{n_{k^*}}$ shows the parts assigned to the robot k^* in terms of the number of alternative parts n_{k^*}.

The n_{k^*} value for the robot k^* is increased by 1 ($n_{k^*} \leftarrow n_{k^*} + 1$).

$$TL_{k^*}^{n_{k^*}} = \max\left(TC_{k^*}, PT_l^{(N_J-1)}\right) + dx_{l,(N_J-1),N_J,m^*,m^{**}}/r \tag{68}$$

If the machine m^* used in operation $(N_J - 1)$ and the machine m^{**} to be used in operation (N_J) are in different cells, the value of $TL_{k+1}^{n_{k+1}}$ is calculated with Equation 69.

$$TL_{k+1}^{n_{k+1}} = \max\left(TC_{k+1}, PT_l^{(N_J-1)}\right) + dy_{l,(N_J-1),N_J,m^*,m^{**}}/f \tag{69}$$

After determining the robots to be used in the relevant operation for all jobs and calculating the $TL_o^{n_o}$ values, the part with the smallest completion time for robot o is found by using Equations 70 and 71.

$$p^* \leftarrow \arg\min_{n_o}\left(TL_o^{n_o}\right) \tag{70}$$

$$\text{part} \leftarrow As_o^{p^*} \tag{71}$$

The transportation completion times of all of the robots TC_o are updated. The completion time of the operation N_{part} of the relevant part on the relevant machine is calculated with Equation 72 and denoted by CT_{m^*}. When calculating CT_{m^*}, the processing start time of the part on machine m^* is taken as $\max(CT_{m^*}, TC_o)$. Thus, processing of the related part begins after transportation for the relevant job is completed and the machine becomes idle. The start time and the processing time are added to determine the completion time (CT_{m^*}).

$$CT_{m^*} \leftarrow \max(CT_{m^*}, TC_o) + P_{part,N_{part},m^*} \tag{72}$$

The operation count of the part is increased by 1. If the last operation of the part has been performed, the NT_{part} value is updated from 0 to 1, and the S value is increased by one. The steps continue until the S value, that is, the part number on which the last operation was performed, is equal to Z. The objective function for the relevant individual is recorded, and the algorithm runs until the objective functions of all individuals are calculated.

Procedure: Scheduling algorithm considering material handling robots

Input: $A_i^{k,l}$,P, $P_{p,s,m}$, Cx_m, Cy_m, G, T,$r, f, a_{p,r,s,m}$,$O_{p,r}$,R_i, $route_i$

Output: Objective function value of solutions $(C_{\max(i)})$

For i=1:P

$CT_m \leftarrow 0 \quad (1 \le m \le T)$

For l=T+1:T+Z

$J \leftarrow R_i^l$; $N_J \leftarrow 1$;

Assign J to related machine m^* of operation N_J considering route;

Calculate completion time of the operation of N_J of part J; $CT_{m^*} \leftarrow CT_{m^*} + P_{J,N_J,m^*}$;

$PT_J^{N_J} \leftarrow CT_{m^*}$; $N_J \leftarrow N_J + 1$;

End

$TC_o \leftarrow 0$; $NT_l \leftarrow 0$; $S \leftarrow 0$; $(1 \le o \le k+1)$ $(1 \le l \le Z)$

While$(S<Z)$

$TL_o^{n_o} \leftarrow 0$; $n_o \leftarrow 1$; $(1 \le o \le k+1)$

For l=1:Z

If $(NT_l == 0)$

Determine the material handling robot to transfer part l between machines of operation N_J and $(N_J - 1)$ and assume the operation of $(N_J - 1)$ is on machine m^* and the operation of N_J is on m^{**};

If m^* and m^{**} are in the same cell an in cell k^* use the material handling robot k^*

$TL_{k^*}^{n_{k^*}} = \max(TC_{k^*}, PT_l^{(N_J-1)}) + dx_{l,(N_J-1),N_J,m^*,m^{**}}/r$; $As_{k^*}^{n_{k^*}} \leftarrow l$; $n_{k^*} \leftarrow n_{k^*} + 1$;

End

If m^* and m^{**} are in different cells use the material handling robot $k+1$;

$TL_{k+1}^{n_{k+1}} = \max(TC_{k+1}, PT_l^{(N_J-1)}) + dy_{l,(N_J-1),N_J,m^*,m^{**}}/f$;

$As_{k+1}^{n_{k+1}} \leftarrow l$; $n_{k+1} \leftarrow n_{k+1} + 1$;

End

End

End

For o=1:k+1

Determine the part with minimum TL_o;

$p^* \leftarrow \arg\min_{n_o}(TL_o^{n_o})$; part$\leftarrow As_o^{p^*}$; $TC_o \leftarrow \min_{n_o}(TL_o^{n_o})$;

Assign part to related machine m^* of operation N_{part} considering route;

Calculate completion time of the operation of N_{part} of part;

$CT_{m^*} \leftarrow \max(CT_{m^*}, TC_o) + P_{part,N_{part},m^*}$; $PT_{part}^{N_{part}} \leftarrow CT_{m^*}$; $N_{part} \leftarrow N_{part} + 1$;

If $(N_{part} > O_{part,r})$ and $(NT_{part} == 0)$

$NT_{part} \leftarrow 1$; $S \leftarrow S+1$;

End

End

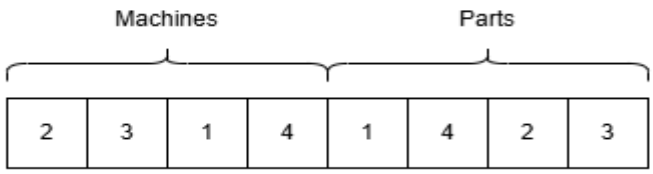
End

$C_{\max(i)} \leftarrow \max(CT_m)$ $(1 \le m \le T)$

End

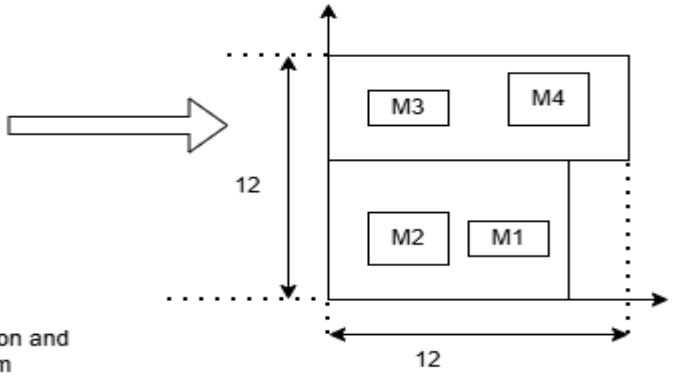
The proposed algorithm is illustrated through an example problem, as shown in Figure 8. In the first stage, a random individual is generated. In the second stage, the cell formation and layout algorithm is applied based on this individual. As a result, Machines 1 and 2 are assigned to Cell 1, while Machines 2 and 3 are assigned to Cell 2. In the third stage, a processing route is determined for each part, either randomly or by selecting the route with minimum transportation cost. In the example problem, random routes are selected for all parts.

In the final stage, scheduling is performed by explicitly considering the material handling robots. According to the obtained results, the objective function value is 29.75, which is equal to the optimal solution of the problem.

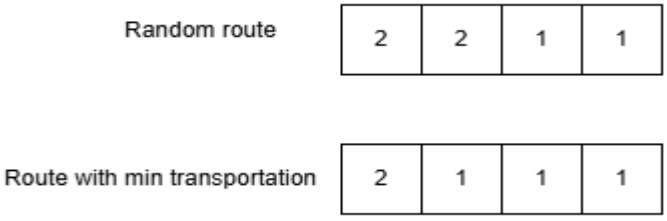


Stage 1: Random Individuals

Machine	k*	S_i^{k*}	W_{k*}
2	1	1	6
3	2	1	7
1	1	2	9
4	2	2	11



Stage 2: Cell formation and layout algorithm



Stage 3: Route selection

Part	Route	Machine	Starting time	Comp. time	Intra cell trans.	Intercell trans.	Trans. start	Trans. comp.
1	2-3-4	M2	0	1	-	✓	1	7.25
		M3	7.25	10.25	✓(R2)	-	10.25	12.75
		M4	12.75	16.75	-	-	-	-
4	2-3	M2	1	2		✓	7.25	13.5
		M3	13.5	16.5	-	-	-	-
2	1-2-3	M1	0	2	✓(R1)	-	2	4
		M2	4	5	-	✓	13.5	19.75
		M3	19.75	22.75	-	-	-	-
3	3-1	M3	0	3	-	✓	19.75	22.75
		M1	22.75	29.75	-	-	-	-

Stage 4: Scheduling algorithm considering the material handling robots.

Figure 8. Application of the decoding algorithm to the illustrative example problem.

5. Analysis

Test problems, the algorithm parameters, comparisons of the proposed algorithm with the GA, HC, and ABC algorithms, and statistical analysis are discussed in this section.

5.1. Test problems

The dataset was generated in accordance with the literature [4,38]. The characteristics of the dataset are provided in Table 2. Instances with 3 machines are small instances, those with 7 or 10 machines are medium instances, and those with 15 or 20 machines are large instances. All instances are represented as Mendeley data [Bektur, Gülçin (2025), "Cell Formation Cell layout and scheduling problem", Mendeley Data, V1, doi: 10.17632/59jy9zpcjz.1].

Table 2 summarizes the main characteristics of the test instances used in the computational study. Structural variability is introduced through different numbers of alternative routes and cell configurations. In particular, the number of routes controls the degree of routing flexibility, whereas the number of cells reflects different levels of intercell movement. This design allows for the performance of the proposed approach to be evaluated across a wide range of problem sizes and structural conditions. Two test problems were generated for each problem characteristic, resulting in a total of 102 test problems.

Table 2. Characteristics of the instances.

# Machines	# Parts	# Routes	# Cells	Size
3	5, 7, 10, 12, or 15	2 or 3	2	Small
7	5, 7, 10, 12, 15, or 20	3	2 or 3	Medium
10	10, 12, 15, 20, or 25	3	3 or 4	Medium
15	12, 15, 20, 25, or 30	3	3 or 4	Large
20	15, 20, 25, 30, or 35	3	3 or 4	Large

5.2. Comparison

In this study, the proposed algorithm was compared with the MILP, GA, ABC, simulated annealing (SA) [11], and HC algorithms. The parameters of the proposed algorithm are $N_{pop}, Mu, Cr, NP, maxIter, limit\ and\ maxtrial$. These parameters are determined on the basis of the literature and preliminary experiments and are provided in Table 3. The HH, ABC, HC, and GA algorithms were run with the same number of iterations and solutions. For SA, the parameters include the initial temperature (T_0), the cooling rate (q), the number of iterations at each temperature (N), and the final temperature (T_f). In the SA algorithm, neighborhood solutions are generated using swap moves applied to the permutation representation. The decoding algorithm is then employed to transform these permutations into feasible solutions and to evaluate their objective function values.

The results of the instances considering the MILP, GA, HH, ABC, HC, and SA algorithms according to the number of machines are given in Tables 4–8. In these tables, T represents the number of machines, N represents the number of cells, Z represents the number of parts, and F represents the number of routes. As shown in Table 4, the MILP model found optimal solutions for instances with 3 machines and 5 or 7 parts. Furthermore, the MILP model obtained a feasible solution to instances with

3 machines and 10, 12, or 15 parts. The proposed algorithm also found optimal solutions for instances in which the MILP model found optimal solutions. However, the proposed algorithm produces the same or better solutions for instances in which the MILP model provided feasible solutions, as is also shown in the table. Instances with 7 machines are given in Table 5, which shows that the MILP model provided a feasible solution to instances with 7 machines and 7, 10, or 12 parts. The HH algorithm provided better solutions than the MILP model. However, as shown in Table 5, the proposed algorithm produces better or the same solutions as the MILP, GA, ABC, HC, and SA algorithms for all instances. The results with 10 machines are presented in Table 6, which shows that the HH algorithm produced better or the same solutions as those provided by the ABC, GA, HC, and SA algorithms except in one instance. In Table 7, the results for 15 machines are given. This table shows that the HH algorithm provided the best solutions for all instances. Furthermore, as shown in Table 8, the HH algorithm produced the same or better results except for one test problem.

Table 3. Parameters of the algorithms.

Algorithms	**Parameters**	**Value**
Hyperheuristic	N_{pop}	200
	Mu/Cr	0.8/0.2
	NP	100
	Δ_{HH}	0.75
	$maxIter$	10
	$limit$	5
	$maxtrial$	100
GA	N_{pop}	200
	Mu/Cr	0.8/0.2
	NP	1000
ABC	$limit$	5
	N_{pop}	200
	$maxtrial$	1000
Hill climbing	N_{pop}	200
	NP	1000
SA	T_0	200
	q	0.99
	N	200
	T_f	0.01

Table 4. Results for test problems with 3 or 7 machines.

T	Z	F	N	MILP			GA		HH		ABC		HC		SA	
				Z		CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
3	5	2	2	**120.25**	*	3	**120.25**	4.17	**120.25**	4.9	**120.25**	7.92	**120.25**	3.61	**120.25**	4.63
3	5	2	2	**77**	*	2	**77**	3.3	**77**	4.94	**77**	6.57	**77**	3.14	**77**	5.18
3	7	2	2	**169**	*	942	**169**	3.64	**169**	5.27	**169**	7.54	**169**	3.48	**169**	5.44
3	7	2	2	**131**	*	92	**131**	3.23	**131**	4.23	**131**	6.65	**131**	3.06	**131**	6.16
3	10	2	2	191		3600	**186**	4.28	**186**	6.17	**186**	9.11	**186**	4.2	**186**	8.21
3	10	2	2	**191**		3600	**191**	4.05	**191**	9.04	**191**	8.51	**191**	3.89	**191**	5.81
3	12	2	2	**247**		3600	**247**	5.05	**247**	19.7	**247**	10.61	**247**	4.89	**247**	8.41
3	12	2	2	**193**		3600	**193**	4.23	**193**	8.92	**193**	8.96	**193**	4.14	**193**	6.45
3	15	2	2	278		3600	**271**	5.84	**271**	12.38	**271**	12.52	276	5.61	276	9.42
3	15	2	2	**247**		3600	**247**	5.97	**247**	13.45	**247**	12.59	**247**	5.8	**247**	12.81
3	5	3	2	**72**	*	7	**72**	3.01	**72**	4.64	**72**	6.22	**72**	2.86	**72**	4.87
3	5	3	2	**68.25**	*	31	**68.25**	2.85	**68.25**	4.09	**68.25**	6.02	**68.25**	2.98	**68.25**	5.99
3	7	3	2	**106**	*	510	**106**	3.43	**106**	3.78	**106**	7.33	**106**	3.34	**106**	6.33
3	7	3	2	**106**	*	921	**106**	3.23	**106**	4.19	**106**	6.86	**106**	3.1	**106**	5.12
3	10	3	2	140		3600	154	4.14	**140**	14.29	**140**	8.89	154	3.97	154	7.98
3	10	3	2	162		3600	**152**	4.33	**152**	9.51	**152**	9.12	157	4.15	157	8.11
3	12	3	2	**160**		3600	172	4.28	**160**	6.7	164	9.13	166	4.11	164	8.13
3	12	3	2	216		3600	202	4.9	**191**	11.24	201	10.54	207	4.78	207	11.71
3	15	3	2	215		3600	221.75	5.28	**207**	14.69	234	12.05	236.25	5.08	234	5.13
3	15	3	2	224		3600	**224**	5.06	**224**	15.56	**224**	11.24	225	4.86	225	8.28
7	5	2	2	**129.75**	*	412	**129.75**	4.13	**129.75**	5.54	**129.75**	8.89	**129.75**	4	**129.75**	6.34
7	5	2	2	**105.75**	*	105	**105.75**	4.27	**105.75**	9.13	**105.75**	8.97	**105.75**	4	**105.75**	7.35

* Optimal solution

Table 5. Results for test problems with 7 machines.

T	Z	F	N	MILP		GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
7	7	3	2	125.5	3600	126.25	6.25	**124.5**	8.66	125.5	12.96	125.5	6.2	125.5	11.74
7	7	3	2	181	3600	**131.5**	5.9	**131.5**	8.91	133	11.99	**131.5**	5.88	133	12.34
7	10	3	2	230.5	3600	**152**	6.92	**152**	8.42	**152**	14.07	**152**	6.88	**152**	13.45
7	10	3	2	-		**125**	6.42	**125**	8.32	**125**	12.99	**125**	6.29	**125**	11.43
7	12	3	2	241.5	3600	153	6.27	**148**	8.97	153.75	13.2	153	6.18	153	12.34
7	12	3	2	-		173	7.53	**166.5**	9	173	15.34	173	7.45	173	173
7	15	3	2	-		**184.5**	9.62	**184.5**	12.52	187.5	19.79	186	9.59	187.5	18.41
7	15	3	2	-		**189.75**	9	**189.75**	9.91	**189.75**	18.55	**189.75**	8.97	**189.75**	17.67
7	20	3	2	-		283	13.2	**275**	17.39	**275**	27.06	282	13.1	282	12.45
7	20	3	2	-		297.75	11.8	**281**	14.53	282.75	24.76	294.25	11.8	**281**	15.32
7	7	3	3	-		**120.25**	6.08	**120.25**	8.13	**120.25**	13	**120.25**	6.07	**120.25**	12.45
7	7	3	3	-		148.75	6.09	**147.5**	6.8	149.75	12.86	**147.5**	6.11	**147.5**	11.35

Continued on next page

T	Z	F	N	MILP		GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
7	10	3	3	-		139	7.52	**134.5**	9.65	135	15.24	137	7.2	137	12.35
7	10	3	3	-		**174.5**	35.4	**174.5**	9.82	**174.5**	19.24	**174.5**	38.4	**174.5**	18.37
7	12	3	3	-		**173**	38.3	**173**	11.9	173.5	19.71	174	11.1	174	18.45
7	12	3	3	-		**194.25**	47.8	**194.25**	11	**194.25**	19.24	**194.25**	8.91	**194.25**	19.56
7	15	3	3	-		**164**	52.4	**164**	13.64	**164**	21.51	**164**	10.2	**164**	22.46
7	15	3	3	-		**141**	49	**141**	14.36	**141**	19.36	**141**	9.28	**141**	21.46
7	20	3	3	-		**240**	51.2	**240**	18.21	**240**	28.24	**240**	13.5	**240**	27.32
7	20	3	3	-		**213.25**	14.2	**213.25**	20.15	214.5	29	**213.25**	14.2	214.5	28

Table 6. Results for test problems with 10 machines.

T	Z	F	N	GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
10	10	3	3	153.25	10.5	**152.25**	14.23	154.25	21.09	154.25	9.91	154.25	19.9
10	10	3	3	**92.25**	8.31	**92.25**	10.16	**92.25**	17.23	**92.25**	8.2	**92.25**	18.25
10	12	3	3	**197.75**	9.62	**197.75**	14.23	204	20.68	198.25	9.68	198.25	19.65
10	12	3	3	**187**	11	**187**	13.54	191	22.66	188	11	191	21.21
10	15	3	3	**195.5**	13.4	**195.5**	19.3	199.5	27.84	197.5	13.3	197.5	23.3
10	15	3	3	198.25	13	191	15.11	**190.75**	27.49	191.25	13.2	191.25	23.28
10	20	3	3	212.5	17.4	**211**	27.1	213	36.17	212.5	17.3	213	37.3
10	20	3	3	254.75	20.2	**253.25**	26.55	**253.25**	45.26	254.75	20.6	254.75	30.68
10	25	3	3	277	21.9	**268**	33.62	277	49.07	277	23.1	277	42.1
10	25	3	3	**212.75**	21.9	**212.75**	26	221	45.01	219.75	22.6	219.75	42.65
10	10	3	4	**153.75**	9.82	**153.75**	14.71	**153.75**	20.34	**153.75**	9.54	**153.75**	19.51
10	10	3	4	**171.5**	12.2	**171.5**	19.42	**171.5**	25.44	**171.5**	11.6	**171.5**	21.69
10	12	3	4	**162.75**	12.7	**162.75**	19.79	**162.75**	26.5	**162.75**	11.9	**162.75**	21.98
10	12	3	4	149.75	10.9	**148**	13.06	155	22.92	156.5	10.2	155	20.28
10	15	3	4	**176**	15.5	**176**	20.51	**176**	32.3	**176**	14.4	**176**	31.48
10	15	3	4	215.5	17.4	**194**	23.06	217.5	35.67	215.5	16.9	215.5	30.94
10	20	3	4	**208.75**	22.9	**208.75**	27.7	211	47.2	215.25	22.1	215.25	40.18
10	20	3	4	223.5	23	**220**	27.17	228	46.35	228	20.9	228	40.41
10	25	3	4	331.25	36.2	**319.25**	50.45	332	67.9	322	31.7	332	61.78
10	25	3	4	247	27.2	**237**	38.9	244	56.47	248	28.1	248	58.18

Table 7. Results for test problems with 15 machines.

T	Z	F	N	GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
15	12	3	3	238.25	26.3	**231.75**	27.45	242.25	34.38	242.25	16.3	242.25	36.6
15	12	3	3	**225.75**	22.9	**225.75**	25.44	**225.75**	35.02	**225.75**	15.9	**225.75**	45.9
15	15	3	3	295.25	26.6	**290.5**	27.44	**290.5**	47.69	297.75	21.8	**290.5**	31.9

Continued on next page

T	Z	F	N	GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
15	15	3	3	**297.5**	23.1	**297**	31.25	301.75	49.08	301.75	22.4	301.75	42.8
15	20	3	3	404.25	37.8	**393.25**	50.86	439.5	81.11	410.75	37.5	439.5	47.5
15	20	3	3	354	32.2	**336.75**	44.58	364.5	70.66	352.75	32.2	364.5	32.41
15	25	3	3	429.25	47	**420**	66.87	462	102.3	427	50.1	427	65.1
15	25	3	3	399	46.7	**395**	66.15	426.5	95.15	416.5	44.3	426.5	64.38
15	30	3	3	407.25	59	**392.5**	70.68	410.5	120.8	412	56.3	412	76.1
15	30	3	3	517.25	65.6	**499**	77.56	507.5	138.8	514.5	63.2	514.5	73.4
15	12	3	4	246	23	**244.75**	25.87	263.5	47	263.5	22.4	263.5	32.5
15	12	3	4	232.25	21.4	**230.5**	25.77	239	43.48	231.75	20.6	239	40.7
15	15	3	4	296	26.6	**285**	31.91	291	53.82	291.5	25.6	291.5	35.5
15	15	3	4	**206.5**	22.3	**206.5**	26.54	209.25	44.98	207	21.3	207	31.8
15	20	3	4	333.5	36.8	**315.75**	47.26	**315.75**	72.95	328	36	328	76.2
15	20	3	4	372.5	38.7	**353.75**	55.09	367.25	78.35	376	37.8	367.25	67.4
15	25	3	4	399.5	56.4	**394.25**	67.44	405	113.6	414	55.2	414	65.8
15	25	3	4	388.75	58.3	**383.5**	64.44	401.75	117.4	419.75	57.6	401.75	87.4
15	30	3	4	398	65.7	**393.5**	100.5	404.5	131.6	404	63.2	404	93.2
15	30	3	4	494.25	78.1	**467**	296.5	516.75	161.1	491.5	77.1	491.5	157.1

Table 8. Results for test problems with 20 machines.

T	Z	F	N	GA		HH		ABC		HC		SA	
				Z	CPU	Z	CPU	Z	CPU	Z	CPU	Z	CPU
20	15	3	3	327.25	41	**326**	56.3	334	81.38	338.5	27.4	334	61.25
20	15	3	3	542.25	50.8	**503.75**	75.96	533	103.1	520.25	34.2	533	85.24
20	20	3	3	419.25	55	**393**	64.79	408.25	111.1	411.5	37.4	411.5	84.21
20	20	3	3	416.5	54.5	**412.75**	71.27	426	112.2	**412.75**	37.3	426	101.52
20	25	3	3	**520.5**	79.6	**520.5**	104	563.75	164.2	557.25	55.3	574.25	125.47
20	25	3	3	682	89.9	**651.75**	126.3	755	185.4	732.5	62.8	741	160.41
20	30	3	3	**629.5**	102	**629.5**	122.1	666.75	210.9	683	71	666.75	188.47
20	30	3	3	577.75	98.6	**558**	142.3	560.5	197.5	622.75	68	611.25	187.32
20	35	3	3	**502**	118	**502**	152.3	503.75	235.2	522.5	83.7	520.12	174.65
20	35	3	3	665.25	129	**654**	156.6	682.25	258.2	685.25	91	682.25	174.21
20	15	3	4	381	46.2	**377**	55.55	382.25	94.05	377.25	32.7	382.25	100.21
20	15	3	4	**313**	46.3	**313**	51.45	**313**	135.5	332	32.6	321	120.74
20	20	3	4	**284**	55.7	**284**	73.74	300.5	182.3	295.25	39.1	300.5	147.41
20	20	3	4	**325**	57.3	329.25	74.33	329.25	185.2	341	40	329.25	165.41
20	25	3	4	552.75	99.7	**529**	123.6	537.75	301.9	571	67.1	537.75	184.25
20	25	3	4	417.5	78.2	**403.5**	101.4	436.75	247.2	420.25	54.8	420.25	169.84
20	30	3	4	**559**	128	**559**	137	591.5	402	578.75	88.3	598.26	198.23
20	30	3	4	561	116	**552.25**	152.6	590	441	565	82.9	587.33	200.41
20	35	3	4	604.25	149	**583.25**	196.5	680.5	372.4	643.75	106	643.75	209.52
20	35	3	4	767.25	159	**756**	175	**756**	232.9	787.25	112	787.25	200.32

The %Error of the algorithms are calculated using Equation 73.

$$\%Error = \frac{Solution\ of\ the\ heuristic\ algorithm - Best\ solution\ found\ by\ heuristics}{Best\ solution\ found\ by\ heuristics} \tag{73}$$

The %Error values of the test problems according to the number of machines are shown in Table 9, which shows that the most successful algorithm is the HH algorithm.

Table 9. %Error values of the algorithms according to the number of machines.

Machine number	HH	GA	ABC	HC	SA
3	**0**	0.0152	0.0104	0.0209	0.0198
7	**0**	0.0099	0.0067	0.0085	0.0075
10	**0.0001**	0.0155	0.0227	0.0202	0.0222
15	**0**	0.0244	0.045	0.0381	0.0415
20	**0.0007**	0.0206	0.0491	0.0538	0.0542

Statistically significant differences were analyzed between the HH algorithm and the other algorithms. The Wilcoxon signed-rank test was selected because it does not assume normality and is well-suited for comparing heuristic algorithms on paired benchmark problems. Pairwise comparisons were conducted between the proposed HH algorithm and each competing method.

The Wilcoxon signed-rank test results are reported in Table 10. The results indicate statistically significant differences between the proposed HH algorithm and all competing methods ($p < 0.001$), and all findings remained significant after Bonferroni correction. Moreover, the effect sizes obtained from the Wilcoxon signed-rank tests were large in all pairwise comparisons ($r > 0.85$). In terms of the commonly accepted thresholds, these values indicate a strong practical impact, confirming that the superiority of the HH algorithm is not only statistically significant but also practically meaningful.

Table 10. Wilcoxon signed-rank test results comparing HH with competing algorithms.

Pairs (HH-X)	p-value	Effect size (r)
(HH-GA)	2.79×10^{-10}	0.85
(HH-ABC)	5.83×10^{-12}	0.87
(HH-HC)	3.80×10^{-13}	0.87
(HH-SA)	3.80×10^{-13}	0.87

5.3. Computational time analysis

The impact of problem dimensions on the computational performance of the heuristic algorithm was examined by varying the numbers of machines, parts, routes, and cells. The results revealed that the solution time increases consistently as the size and structural complexity of the problem increase. In particular, larger sets of parts and machines lead to greater computational effort, whereas additional routes and cells further expand the decision space and thus prolong the search process. Nevertheless, the observed runtime growth remains gradual rather than abrupt, and even the largest tested instances can be solved within acceptable time limits. This demonstrates that the proposed heuristic maintains

favorable scalability with respect to both the scale and the structural parameters of the assignment–scheduling problem.

To determine which problem parameters have the strongest impact on computational time, a multiple regression analysis was performed using runtime as the dependent variable and the number of machines, parts, routes, and cells as explanatory variables. This approach allows for the individual effect of each parameter to be evaluated while controlling for the others. Statistical significance was assessed at the 95% confidence level, and the relative importance of factors was interpreted on the basis of the magnitude of the estimated coefficients and their associated t-statistics.

The regression results are given in Table 11. The results reveal that the number of parts is the dominant factor affecting the solution time, with the largest and most significant coefficient. The number of machines also has a positive and statistically significant effect, although its influence is relatively small. The intercept is reported for completeness; however, it does not have a direct practical meaning for the problem under study and is therefore not discussed.

Table 11. The regression results.

Variable	Coefficient (β)	Standard error	t-statistic	p-value
Constant	7.4485	24.926	0.299	0.766
Machines	3.4108	0.749	4.552	<0.001
Parts	3.8792	0.470	8.258	<0.001
Routes	−20.0434	9.426	−2.127	0.036
Cells	−3.0706	4.843	−0.634	0.528

5.4. Sensitivity analysis

To evaluate the robustness of the proposed HH algorithm, a parameter sensitivity study was performed. The analysis focused on two key control factors, maxIter and Δ_{HH}, to investigate how different parameter configurations affect the algorithmic behavior. maxIter was set to (7;10;15), and Δ_{HH} was set to (0,25; 0,50; 0,75). A full factorial experimental framework was adopted so that all combinations of the selected parameter levels could be examined in a systematic manner. Performance was assessed using the relative percentage deviation (RPD), which is defined in Eq. (74)

$$RPD_{t,k,r} = \frac{OF_{t,k,r} - OFmin_{k,r}}{OFmin_{k,r}} \tag{74}$$

$OF_{t,k,r}$ denotes the objective value obtained at time step t for instance k in replication r, and $OFmin_{k,r}$ represents the best value observed for the same instance. The statistical significance of the parameter effects was examined using analysis of variance (ANOVA). The resulting interval plots, which illustrate the impact of different parameter settings, are reported in Figure 9.

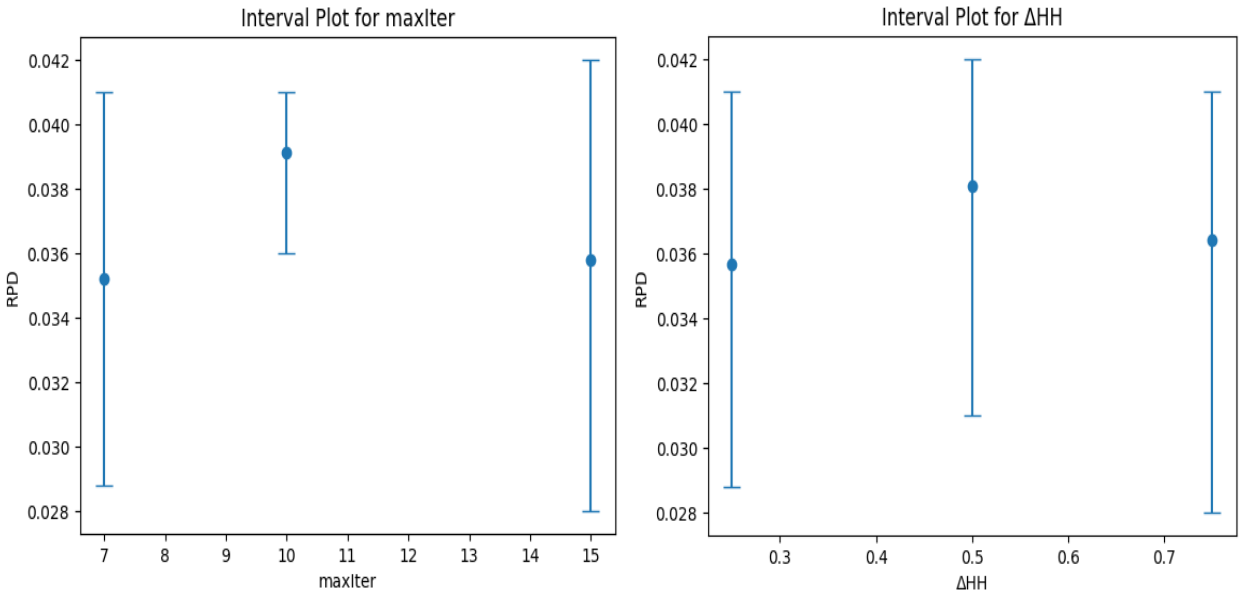


Figure 9. Interval plots for maxIter and Δ_{HH}.

The analysis revealed that both main effects were statistically significant ($p < 0.05$), indicating that each parameter individually influences performance. In addition, the interaction term maxIter × Δ_{HH} was highly significant, revealing that the effect of one parameter depends on the level of the other. This outcome is consistent with the interval plots, which highlighted a stable operating region around maxIter = 10 where variations in Δ_{HH} led to only modest changes in the RPD. Overall, although the parameters are statistically influential, their combined impact remains within a narrow range.

5.5. Practical realism and limitations

The proposed framework aims to reflect realistic CMS settings by explicitly modeling a robot-based material handling system within the integrated CF–CL–scheduling problem. The assumption of one robot per cell and a single robot for intercell transportation represents common automated CMS configurations and enables the modeling of robot availability, nonoverlapping transport operations, and waiting times because of limited handling capacity.

Practical realism is further enhanced by distinguishing between intracell transportation and assigning different travel speeds to these operations. However, to preserve tractability, identical robot characteristics are assumed within each cell. Allowing for heterogeneous intracell robot speeds would define a more generalized and significantly more complex problem in which robot speeds could directly affect part-to-cell assignments. Even under the current assumption, the problem remains highly challenging because each robot can handle only one part at a time.

With respect to exact optimization, the MILP model provides optimal solutions for small-sized instances (e.g., 7 machines and 5 parts) and only feasible solutions for medium-sized instances (e.g., 7 machines and 12 parts) within a 3600-s time limit. For larger instances, no feasible solutions can be obtained within this limit, which highlights the scalability limitations of the MILP approach. Consequently, the MILP model was used as a benchmark for small instances, whereas the proposed HH can effectively address large-scale problems. Potential improvements in the mathematical formulation

using decomposition or acceleration techniques such as second-order cone programming [39] may be considered in future studies.

6. Conclusions

In this study, an integrated CF, CL, and scheduling problem with material handling constraints and different routes is addressed. Although the CF, CL, and scheduling problems are addressed in the literature, the material handling constraints of parts in the same and different cells are often ignored. Unlike in the literature, in this study, the integrated problem is handled by considering the material handling system and alternative routes. An MILP model is proposed to obtain optimal solutions, and a HH algorithm that incorporates a decoding algorithm is proposed. The proposed algorithm was compared with the MILP model and the GA, ABC, and HC algorithms. According to the results, the HH algorithm is effective. In future studies, different objective functions may be considered, and the problem will be modified for multi-objective optimization. In addition, exact-solution approaches such as decomposition may be used to obtain optimal solutions. In addition, different heuristic algorithms for the problem may be proposed in future research.

Use of Generative-AI tools declaration

The authors declare that they did not utilize any artificial intelligence (AI) tools in the creation of this article.

Author contributions

Gulcin Bektur: Methodology, conceptualization, supervision, software, validation, visualization, writing- original draft. **Mehmet Palta:** Software, validation, writing- original draft.

Conflict of interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. X. Wu, C. Chu, Y. Wang, D. Yue, Genetic algorithms for integrating cell formation with machine layout and scheduling, *Comput. Ind. Eng.,* **53** (2007), 277–289. https://doi.org/10.1016/j.cie.2007.06.021
2. R. Motahari, Z. Alavifar, A. Z. Andaryan, M. Chipulu, M. Saberi M, A multi- objective linear programming model for scheduling part families and designing a group layout in cellular manufacturing systems, *Comput. Oper. Res.*, **151** (2023), 106090. https://doi.org/10.1016/j.cor.2022.106090
3. V. Rahimi, J. Arkat, H. Farughi, A vibration damping optimization algorithm for the integrated problem of cell formation, cellular scheduling, and intercellular layout, *Comput. Ind. Eng.*, **143** (2020), 106439. https://doi.org/10.1016/j.cie.2020.106439

4. S. Vitayasak, P. Pongcharoen, Performance improvement of Teaching- Learning- Based optimization for robust machine layout design, *Expert Syst. Appl.*, **98** (2018), 129–152. https://doi.org/10.1016/j.eswa.2018.01.005
5. H. Chen, P. Wang, J. Li, G. Zhang, Y. Zhang, An improved NSGA II- SA algorithm for the cell manufacturing system layout optimization problem, *Oper. Res.,* **25** (2025), 1–31. https://doi.org/10.1007/s12351-025-00899-0
6. K. Forghani, S. M. T. Ghomi, Joint cell formation, cell scheduling, and group layout problem in virtual and classical cellular manufacturing systems, *Appl. Soft Comput.*, **97** (2020), 106719. https://doi.org/10.1016/j.asoc.2020.106719
7. A. Ebrahimi, R. Kia, A. R. Komijan, Solving a mathematical model integrating unequal- area facilities layout and part scheduling in a cellular manufacturing system by a genetic algorithm, *SpringerPlus,* **5** (2016), 1254. https://doi.org/10.1186/s40064-016-2773-5
8. S. Fahmy, Mixed integer linear programming model for integrating cell formation, group layout and group scheduling, *IEEE International Conference on Industrial Technology, (2015)*, *ICIT*, *IEEE* 2403–2408. https://doi.org/10.1109/ICIT.2015.7125452
9. J. Arkat, M. H. Farahani, L. Hosseini, Integrating cell formation with cellular layout and operations scheduling, *Int. J. Adv. Manuf. Technol.*, **61** (2012), 637–647. https://doi.org/10.1007/s00170-011-3733-4
10. J. Arkat, M. H. Farahani, F. Ahmadizar, Multi- objective genetic algorithm for cell formation problem considering cellular layout and operations scheduling, *Int. J. Comput. Integr. Manuf.,* **25** (2021), 625–635. https://doi.org/10.1080/0951192X.2012.665182
11. H. Heydari, M. M. Paydar, I. Mahdavi, A novel hybrid approach for designing green robust manufacturing cells, *Comput. Ind. Eng.*, **201** (2025), 110946. https://doi.org/10.1016/j.cie.2025.110946
12. H. Bayram, R. Sahin, A comprehensive mathematical model for dynamic cellular manufacturing systems design and linear programming embedded hybrid solution techniques, *Comput. Ind. Eng.*, **91** (2016), 10–29. https://doi.org/10.1016/j.cie.2015.10.014
13. K. Forghani, S. M. T. Ghomi, R. Kia, Group layout design of manufacturing cells incorporating assembly and energy aspects, *Eng. Optim.*, **54** (2022), 770–785. https://doi.org/10.1080/0305215X.2021.1900155
14. K. Chandrasekar, P. Venkumar, A simulated annealing approach for integrating cell formation with machine layout and cell layout, *Int. Robot. Autom.*, **28** (2013). https://doi.org/10.2316/Journal.206.2013.3.206-3917
15. H. Feng, L. Xi, T. Xia, E. Pan, Concurrent cell formation and layout design based on hybrid approaches, *Appl. Soft Comput.*, **66** (2018), 346–359. https://doi.org/10.1016/j.asoc.2018.02.021
16. K. Forghani, M. Mohammadi, V. Ghezavati, Integrated cell formation and layout problem considering multi- row machine arrangement and continuous cell layout with aisle distance, *Int. J. Adv. Manuf. Technol.*, **78** (2015), 687–705. https://doi.org/10.1007/s00170-014-6652-3
17. K. Forghani, S. M. T. Ghomi, R. Kia, Solving an integrated cell formation and group layout problem using a simulated annealing enhanced by linear programming, *Soft Comput.*, **24** (2020), 11621–11639. https://doi.org/10.1007/s00500-019-04626-8
18. I. Mahdavi, B. Shirazi, M. M. Paydar, A flow matrix- based heuristic algorithm for cell formation and layout design in cellular manufacturing system, *Int. J. Adv. Manuf. Technol.*, **39** (2008), 943–953. https://doi.org/10.1007/s00170-007-1274-7

19. S. Salimpour, H. Pourvaziri, A. Azab, Semi- robust layout design for cellular manufacturing in a dynamic environment, *Comput. Oper. Res.*, **133** (2021), 105367. https://doi.org/10.1016/j.cor.2021.105367
20. M. S. Jabal-Ameli, M. Moshref-Javadi, B. B. Tabrizi, M. Mohammadi, Cell formation and layout design with alternative routing: a multi- objective scatter search approach, *Int. J. Ind. Syst. Eng.*, **14** (2013), 269–295. https://doi.org/10.1504/IJISE.2013.054281
21. M. Alimian, V. Ghezavati, R. Tavakkoli-Moghaddam, New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems, *J. Manuf. Syst.*, **56** (2020), 341–358. https://doi.org/10.1016/j.jmsy.2020.06.011
22. V. Ghezavati, M. Saidi-Mehrabad, Designing integrated cellular manufacturing systems with scheduling considering stochastic processing time, *Int. J. Adv. Manuf. Technol.*, **48** (2010), 701–717. https://doi.org/10.1007/s00170-009-2322-2
23. S. Karthikeyan, M. Saravanan, K. Ganesh, GT machine cell formation problem in scheduling for cellular manufacturing systems using meta- heuristic method, *Procedia Eng.*, **38** (2012), 2537–2547. https://doi.org/10.1016/j.proeng.2012.06.299
24. C. Liu, J. Wang, J. Leung, K. Li, Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm, *Int. J. Prod. Res.*, **54** (2016), 923–944. https://doi.org/10.1080/00207543.2015.1113328
25. G. Papaioannou, J. M. Wilson, Fuzzy extensions to integer programming models of cell-formation problems in machine scheduling, *Ann. Oper. Res.*, **166** (2009), 163–181. https://doi.org/10.1007/s10479-008-0423-1
26. H. Rafiei, M. Rabbani, H. Gholizadeh, H. Dashti, A novel hybrid SA/ GA algorithm for solving an integrated cell formation- job scheduling problem with sequence- dependent set- up times, *Int. J. of Manag. Sci. Eng. Manag.*, **11** (2016), 1–9. https://doi.org/10.1080/17509653.2014.1003109
27. X. Wang, J. Tang, K. Yung, A scatter search approach with dispatching rules for a joint decision of cell formation and parts scheduling in batches, *Int. J. Prod. Res.*, **48** (2010), 3513–3534. https://doi.org/10.1080/00207540902922828
28. A. Delgoshaei, A. Ali, M. K. A. Ariffin, C. Gomes, A multi- period scheduling of dynamic cellular manufacturing systems in the presence of cost uncertainty, *Comput. Ind. Eng.*, **100** (2016), 110–132. https://doi.org/10.1016/j.cie.2016.08.010
29. P. Renna, S. Materi, M. Ambrico, Review of responsiveness and sustainable concepts in cellular manufacturing systems, *Appl. Sci.*, **13** (2023), 1125. https://doi.org/10.3390/app13021125
30. N. Almasarwah, E. Abdelall, M. K. S. Bhutta, M. Saraireh, Designing a cellular manufacturing system as a step toward creating sustainable cells under uncertain demand condition, *J. Remanuf.*, **15** (2025), 151–177. https://doi.org/10.1007/s13243-025-00151-0
31. I. Golcuk, F. B. Ozsoydan, Q-learning and hyper- heuristic based algorithm recommendation for changing environments, *Eng. Appl. Artif. Intell.*, **102** (2021), 104284. https://doi.org/10.1016/j.engappai.2021.104284
32. M. Chen, J. Xu, W. Zhang, Z. Li, A new customer- oriented multi- task scheduling model for cloud manufacturing considering available periods of services using an improved hyper- heuristic algorithm, *Expert Syst. Appl.*, **269** (2025), 126419. https://doi.org/10.1016/j.eswa.2025.126419
33. D. Karaboga, An idea based on honey bee swarm for numerical optimization: Technical report (2005), Erciyes University.

34. J. Chen, Q. Pan, J. Neufeld, Z. Miao, Optimization of task assignment for multi- farm multi-weeding robots based on discrete artificial bee colony algorithm, *Expert Syst. Appl.*, **267** (2025), 126182. https://doi.org/10.1016/j.eswa.2024.126182
35. E. D. Salehi, M. Fazli, Late acceptance hill climbing based algorithm for unmanned aerial vehicles path planning problem, *Appl. Soft Comput.*, **170** (2025), 112651. https://doi.org/10.1016/j.asoc.2024.112651
36. S. Sun, S. Ding, A. Wang, Y. Ding, C. Wei, L. Zhu, et al., An efficient heuristic power analysis framework based on hill-climbing algorithm, *Inf. Sci.*, **662** (2024), 120226. https://doi.org/10.1016/j.ins.2024.120226
37. H. Wang, D. Wang, S. Yang, A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems, *Soft Comput.*, **13** (2009), 763–780. https://doi.org/10.1007/s00500-008-0347-3
38. H. Bouaziz, M. Berghida, A. Lemouari, Solving the generalized cubic cell formation problem using discrete flower pollination algorithm, *Expert Syst. Appl.*, **150** (2020), 113345. https://doi.org/10.1016/j.eswa.2020.113345
39. P. Du, B. Li, Q. Zeng, D. Zhai, D. Zhou, L. Ran, Distributionally robust two- stage energy management for hybrid energy powered cellular networks, *IEEE Trans. Veh. Technol.*, **69** (2020), 1262–1274. https://doi.org/10.1109/TVT.2020.3013877

AIMS Press