Production system configuration and layout planning for efficient manufacturing system design: An industrial case study

Hiroyuki Sakata[a], Péter Dobrovoczki[b], Daisuke Tsutsumi[a], András Kovács[b,*]

[a] *Hitachi Ltd., Research and Development Group, 292 Yoshidacyo, Totsuka, Yokohama, Kanagawa, 244-0817, Japan*
[b] *HUN-REN Institute for Computer Science and Control (SZTAKI), Kende u. 13-17, H-1111 Budapest, Hungary*

Abstract

In recent years, the manufacturing industry has faced increasing challenges related to labor shortages and rising labor costs. One response to these challenges is the automation of production systems, which replaces part of the human workforce with equipment such as robots, machine tools, and conveyors. However, designing an automated production system remains a complex task, often requiring engineers to manually develop an optimal system configuration and layout that minimizes investment costs while satisfying constraints such as production demand, technological requirements, and limited floor space. The traditional approach, solving system configuration and layout planning separately, often requires numerous iterations when floor space is restricted, making it difficult to obtain feasible solutions within a practical time frame. To address this issue, this study applies a recent logic-based Benders decomposition approach to a real industrial production system configuration and layout planning problem, involving the design of a machining cell composed of robots, machines, and other resources. The recently proposed abstract model is extended to capture all practically relevant requirements, including detailed modeling of resources and manufacturing processes. A case study demonstrates how the optimization software can be integrated into the overall planning workflow and highlights the refine-

*Corresponding author
Email address: andras.kovacs@sztaki.hu (András Kovács)

ments made by human experts to adapt the automatically computed solution to fulfill all practical requirements. The results show that, compared to the conventional manual workflow, the proposed optimization approach and software tool reduced the required human design effort from 22 working hours to 4.5 hours.

Keywords: Cellular manufacturing systems, System configuration, Cell formation, Intra-cell layout planning, Benders decomposition, Mixed-integer linear programming

---

1. Introduction

In the manufacturing industry, automation of production systems is required due to several factors, such as growing labor shortages [1] and increasing labor costs [2]. The automation of production systems not only addresses the above mentioned workforce issues but also enhances productivity [3] and flexibility [4] in production processes, making it a crucial aspect for the competitive advantage of manufacturing companies [5]. In order to build an automated production system, manufacturing companies need to execute the engineering tasks such as production system design, manufacturing or procurement of production equipment (e.g., robots, conveyors, and jigs), software implementation, overall production system installation and testing [6, 7]. Production system design includes the following two main subproblems: (i) the system configuration problem, where engineers need to select equipment from the available resources, combine the selected equipment into work cells, and assign products to the selected resources [8]; (ii) the layout planning problem [9], where engineers determine the geometrical position and orientation of the given resources within the available floor space. It is a demanding task for engineers to manually find a system configuration and layout that minimizes investment cost while satisfying complex constraints such as specifications of product mix and available resources, demand volumes and process sequences of each product, as well as the limitations of the available floor space.

In order to support engineers throughout the system design process, several studies proposed optimization approaches separately for the system configuration problem [10] and for the layout planning problem [9]. These approaches have the potential to increase the speed and the quality of the production system design process, and allow engineers to achieve better solutions in shorter time. However, the above approaches overlook the integration of the two related problems, and as a result, the computed system configuration may not fit into the available floor space. In practice, such issues are managed via iterative modifications until a feasible solution is found. While most contributions in the scientific literature focus on greenfield projects, which involve building new systems from scratch, industry more commonly faces brownfield projects, addressing the modification of existing systems [11]. In such cases, the available floor space is strictly limited, which may easily result in numerous iterations, increasing the difficulty of finding a feasible solution within a practical time frame.

An approach for overcoming these difficulties is to integrate the system configuration and the layout planning problems. Yet, integration presents a computational challenge, since both sub-problems are NP-hard in themselves. Logic-based Benders decomposition [12] is an efficient approach for solving such large-scale optimization problems by dividing them into a master problem and one or multiple subproblems. It solves the master problem first, and then each subproblem separately within the frames defined by the master solution. Upon a potential infeasibility of a subproblem, constraints (so-called Benders cuts) are fed back to the master problem for avoiding similar sources of infeasibility in future iterations. The performance of the approach depends heavily on the degree to which decomposition can exploit problem structure, i.e., both on the problem and on the solution techniques (e.g., logical inference techniques, such as lifted cuts) applied within the decomposition framework [12].

A logic-based Benders decomposition approach to generic configuration-and-layout problems was proposed recently in [13]. While that study demonstrated the computational efficiency of the Benders approach on a clear, abstract mathematical model, for successful industrial applications, the problem model must be extended with a number of side constraints that capture detailed practical requirements. Incorporating these application-specific constraints also makes the problem more difficult to solve. Yet, capturing these requirements is an absolute must for bridging the gap between theoretical models and practical applications.

This paper addresses the aforementioned challenge by making the following contributions:

- Extending the abstract configuration-and-layout model of [13] to capture various practical requirements arising in the industrial application.

This includes the detailed modeling of the resources (e.g., robots, machine tools, robot grippers, part stockers, jig stockers, etc.) and the manufacturing process (e.g., changeover times for jigs and robot grippers, part loading and unloading times) on both the configuration and the layout levels; as wells as adding a secondary objective to the layout planning problem to capture the robot travel distance.

- Presenting a detailed case study about how the proposed automated planning approach could be used by a human expert in an actual industrial project, including how it was fitted into the overall planning workflow, what refinements of the computed solutions were needed to respond to all practical requirements, and most importantly, how the automated approach led to a significant decrease of the planning effort required from the human expert.

2. Literature review

2.1. Production system configuration

Optimization of production system configuration has long been a topic of interest for improving productivity and flexibility in the manufacturing industry. A typical example of the system configuration problem is to determine the optimal combination of production resources that minimizes investment cost while satisfying constraints such as product mix, demand volume, and process sequence. From the architectural point of view, two common types of production systems are flow-line systems and cellular manufacturing systems. A flow-line system consists of a sequence of workstations through which products progress in a single direction during the manufacturing process [14]. The cellular manufacturing systems improve production efficiency and flexibility by grouping similar parts into part families and corresponding machines into cells [15].

For flow-line systems, a classical system configuration model is the assembly line balancing (ALB) problem [16]. ALB involves distributing the total workload required to manufacture each unit of product among multiple workstations along the line [10, 17]. Various computational approaches have been proposed for solving ALB problems, including mixed-integer linear programming (MILP) models [18], zero-one integer programming models [19], custom cutting plane algorithms [20], and a graph theory-based approach [21]. However, since ALB is NP-hard, it is difficult to apply exact solution methods to large-scale problems. For this reason, different (meta-)heuristics have also been investigated: e.g., rule-based genetic algorithm (GA) [22], ant colony optimization [23], and particle swarm optimization [24]. [25] proposes a heuristic method for simultaneously solving the task sequencing and the system configuration problem.

For cellular manufacturing systems, the system configuration problem corresponds to cell formation (CF) [26]. A typical CF problem involves determining the number of cells, assigning machines to cells, and allocating parts to machines. Solution methods include MILP [27], tabu search [28], GA and a meta-heuristic called multi-objective vibration damping optimization [29], as well as multi-objective simulation optimization [30].

2.2. Layout planning

Facility layout planning (FLP) involves design problems related to the spatial arrangement of the resources that constitute an industrial production system [9]. A recent literature review on FLP, together with a proposal of research directions to achieve practicable automated layout planning methods, is presented in [31]. Based on the material handling system, FLP problems are typically classified into six categories [9, 32]: single-row layout, double-row layout, parallel-row layout, multi-row layout, loop layout, and open-field layout problems. Specifically, open-field layouts are distinguished as arrangements that do not follow any layout templates.

Various approaches have been proposed for FLP using different layout templates. In [33], robust single row layouts are introduced that tolerate demand uncertainty. [34] proposed a hybrid firefly-chaotic simulated annealing approach to optimize U-shaped single-row layouts. For multi-row layouts, [35] presents a robust machine layout design tool to minimize material flow distance using a GA, taking into account demand uncertainty and machine maintenance. [36] introduced an integer linear programming model for FLP that places a set of fixed-size rectangular departments in such a way that the material flow between adjacent departments is maximized. For loop layouts, several meta-heuristic approaches were investigated, such as a harmony search [37] and a random-key and cuckoo search-based approach [38]. The above models, relying on layout templates, offer the advantage of computational efficiency but are constrained by limited design flexibility.

Various contributions address FLP without layout templates, i.e., the 2D open-field layout problem. A MILP formulation is introduced in [39] for FLP considering material handling points and material path design. MILP models for the dynamic layout case and for a multi-floor variant are presented in [40] and [41], respectively. [42] proposed a three-stage mathematical programming method to find competitive solutions for multi-floor problems. [43] modeled a facility layout problem with conflicting objectives as a Bertrand competition game and solved it using a simulated annealing (SA) meta-heuristic.

2.3. Integration of system configuration and layout planning

Several exact solution approaches have been proposed to find an optimal solution of integrated system configuration and layout planning problems in cellular manufacturing systems. [44] proposed a mixed-integer nonlinear programming model, and demonstrated that the integrated approach outperforms the sequential approach. [45] introduced a MILP model for integrated CF, group layout, and group scheduling. For tackling larger problems, several (meta-)heuristic methods have been investigated. Among others, these include a hybrid GA-SA method for integrated cell formation and layout planning under supply chain uncertainty [46]; a GA to solve cell formation, group layout, and group scheduling [47]; an iterative heuristic for the integrated layout design and product flow assignment problem [48]; and a SA meta-heuristic for CF and group layout in dynamic environments [49].

3. Problem definition

In this chapter we describe the system configuration problem (Section 3.1) and the layout planning problem (Section 3.2). Figure 1 shows the structure of the integrated problem as well as the applied solution method.

3.1. System configuration problem

The cellular manufacturing system configuration problem addressed in this study involves determining the assignment of products to resources and resources to manufacturing cells, while minimizing the investment cost. The system designed in this study is a multi-product production system consisting of multiple cells with deterministic product demand. All products must be produced within the production period without allowing for stock or backlogs.

Each cell consists of one robot, multiple machines, and sub-resources, such as part stocker, gripper stocker, jig stocker and adjustment device. Based on a given process plan, the machine performs a complete machining operation, such as milling and drilling, in a single operation. The robot, equipped with product-specific gripper, performs the loading and unloading of products into and out of the machine. If a cell is assigned more than one robot gripper, then it is necessary to assign as many gripper stockers to the cell as robot grippers. Each cell must have one part stocker for storing pre- and post-processed parts. Each cell must be equipped with one jig stocker, as well as one adjustment device to temporarily store and adjust the part grasp position for the robot. The maximum number of machines that can be installed in a cell is bounded from above.

The resources vary in size, cost, and performance. A machining center with higher cost tends to have shorter machining times and be able to process more products, yet, this is not an assumption in the proposed model. In addition, the robots, machines, and grippers that can be applied to a product are limited by the characteristics of the products (weight, shaft length, shaft diameter, etc.). Such technical requirements are captured via J_p, the set of robot types applicable for product p; N_p, the set of machine types for p; and G_p, the set of gripper types for p.

It is assumed that the products assigned to each machine are processed in a single production batch, e.g., 20 pieces of a product assigned to a machine must be processed consecutively. The operation times required to process are classified as follows.

- **Jig change time** ($T^J_{p,n,r}$) : Time required to change the jig for processing a product, depending on product p, machine type n and robot type r, occupying both the machine and the robot.

- **Loading time** ($T^L_{p,r}$) : Time required to transport each piece of a product to the machine, depending on product p and robot type r, occupying both the machine and the robot.

- **Process time** ($T^P_{p,n}$) : Time required to process a product, depending on product p and machine type n, occupying only the machine.

- **Unloading time** ($T^U_{p,r}$) : Time required to transport each piece of product p from the machine, depending on product p and robot type r occupying both the machine and the robot.

- **Gripper change time** ($T^G_{p,r}$) : the time required before performing the loading and unloading tasks when the robot operates with different grippers, depending on product p and robot type r, occupying only the robot.

The proposed model does not allow for detailed scheduling, thus cannot calculate the exact number of gripper exchanges required. Therefore, it is assumed that a gripper exchange is always required before loading and unloading operations in a multi-gripper cell. This estimation is based on the actual machining cell system and is considered reasonable, since the gripper exchange time is sufficiently short relative to the process time.

The available floor space of the cells are known a priori. In this system configuration problem, only the total floor area of each cell is considered and no layout planning based on geometry is performed. The notation used for the system configuration problem is summarized in Table 1.

Table 1: Notation for system configuration problem.

Indices	
p	Product (index)
c	Cell (index)
r	Robot type (index)
m	Machine slot (index)
n	Machine type (index)
g	Gripper type (index)
i	Gripper stocker type (index)
j	Jig stocker type (index)
k	Part stocker type (index)
a	Adjustment device type (index)
Input parameters	
D_p	Demand for product p [pcs]
J_p	Set of applicable robot types for product p (index set)
N_p	Set of applicable machine types for product p (index set)
G_p	Set of applicable gripper types for product p (index set)
K_r^R	Purchase cost for robot type r [$]
K_n^M	Purchase cost for machine type n [$]
K_g^G	Purchase cost for gripper type g [$]
K_i^{GS}	Purchase cost for gripper stocker type i [$]
K_j^{JS}	Purchase cost for jig stocker type j [$]
K_k^{PS}	Purchase cost for part stocker type k [$]

K_a^{AD}	Purchase cost for adjustment device type a [$]
$\overline{M}_r$	Maximum number of available machine slots if robot r is assigned to the cell [num]
Θ	Length of production period [sec]
D	Length of depreciation period for resources [years]
$T_{p,n}^P$	Process time for product p with machine type n [min]
$T_{p,r}^L$	Part loading time for product p with robot type r [min]
$T_{p,r}^U$	Part unloading time for product p with robot type r [min]
$T_{p,n,r}^J$	Jig change time for product p and machine type n with robot type r [min]
$T_{p,r}^G$	Gripper change time for product p with robot type r [min]
S_r^R	Floor space required by robot type r [m^2]
S_n^M	Floor space required by machine type n [m^2]
S_i^{GS}	Floor space required by gripper stocker type i [m^2]
S_j^{JS}	Floor space required by jig stocker type j [m^2]
S_k^{PS}	Floor space required by part stocker type k [m^2]
S_a^{AD}	Floor space required by adjustment device type a [m^2]
$\overline{S}_c$	Available floor space of cell c [m^2]
Decision variables	
$y_{p,c,r,m,n}$	Denotes if product p is assigned to cell c, machine slot m, to be produced with robot r and machine type n (binary)
$x_{p,c,r,m,n}$	Fraction of the demand for product p assigned to cell c, machine slot m, to be produced with robot type r and machine type n (real in $[0,1]$)
$b_{c,r}$	Cell c is built with robot type r (binary)
$d_{c,m,n}$	In cell c, machine slot m is built with machine type n (binary)
q_c	Cell c is a multi-gripper cell (binary)
t_c^G	Total gripper change time for cell c [min]
$z_{c,g}^G$	Cell c is equipped with gripper type g (binary)
$z_{c,i}^{GS}$	Cell c is equipped with gripper stocker type i (binary)
$z_{c,j}^{JS}$	Cell c is equipped with jig stocker type j (binary)
$z_{c,k}^{PS}$	Cell c is equipped with part stocker type k (binary)
$z_{c,a}^{AD}$	Cell c is equipped with adjustment device type a (binary)

3.2. Layout planning problem

The layout planning problem considers the physical arrangement of ‘items’ such as machines, robots and different types of stockers in order to fit inside the available floor space of each cell. To improve handling efficiency, the objective of the layout planning problem in this study is to minimize the robot travel distance for the robot handling tasks such as part loading and unloading, jig change, and hand change. The available floor space of a cell is given with the width and height of the cell. Other inputs to the layout planning problem are the set of items assigned to the cell, and their floor space requirements. Cells and items are modeled as axis-aligned rectangles, rotations of 90 degrees are allowed for items. A margin is prescribed between resources as a space for human maintenance work in case of resource failure or stoppage during operation. The notation applied for the layout planning problem is summarized in Table 2.

Table 2: Notation for layout planning model.

Indices	
i	Item (index)
c	Cell (index)
Input parameters	
$\mathcal{I}_c$	Items assigned to cell c (set)
$\mathcal{R}_c$	Set of robots assigned to cell c (set)
$\mathcal{M}_c$	Set of machines assigned to cell c (set)
$\mathcal{PS}_c$	Set of part stockers assigned to cell c (set)
$\mathcal{AD}_c$	Set of adjustment devices assigned to cell c (set)
$\mathcal{JS}_c$	Set of jig stockers assigned to cell c (set)
$\mathcal{GS}_c$	Set of gripper stockers assigned to cell c (set)
$M_{i,j}$	Margin required between item i and j (real) [m]
W_i	Width of item i (real) [m]
H_i	Height of item i (real) [m]
$\overline{W}_c$	Width of cell c (real) [m]
$\overline{H}_c$	Height of cell c (real) [m]
Decision variables	
x_i	Coordinate x of the midpoint of item i (real) [m]
y_i	Coordinate y of the midpoint of item i (real) [m]
z_i	Whether item i is rotated 90 degrees (binary)
w_i	Width of item i after considering rotation state (real) [m]
h_i	Height of item i after considering rotation state (real) [m]
$\alpha_{i,j}$	Whether item i located above item j (binary)
$\rho_{i,j}$	Whether item i located on the right of item j (binary)

$d_{i,j}$	Distance of item i and item j (real), [m]
$\delta^x_{i,j}$	Distance of item i and item j along axis x (real) [m]
$\delta^y_{i,j}$	Distance of item i and item j along axis y (real) [m]
d^L	Total robot travel distance for the part loading task (real) [m]
d^U	Total robot travel distance for the part unloading task (real) [m]
d^J	Total robot travel distance for the jig change task (real) [m]
d^G	Total robot travel distance for the gripper change task (real) [m]

4. Solution approach

The system configuration and layout planning problems are modeled separately as MILP, described in Sections 4.1 and 4.2, respectively. The integration of the problems is based on the following iterative procedure, as shown in Figure 1. First, the optimal solution to the system configuration problem is obtained (Figure 1a). Since the geometrical requirements of the layout planning problem are not encoded in the system configuration problem, there is no guarantee that the optimal configuration can be realized in the physical sense. Hence, with the optimal solution of the system configuration as input, the layout planning problem is solved (Figure 1b). If a feasible solution to the layout planning problem is found, the process is terminated. Otherwise, the system configuration problem is extended with new constraints, i.e., Benders cuts, restricting the set of items that can be assigned to a cell. Further discussion of the cuts is given in Section 4.3. Then the system configuration problem is solved again, repeating the cycle until the system configuration admits a feasible layout.

4.1. MILP model for system configuration problem

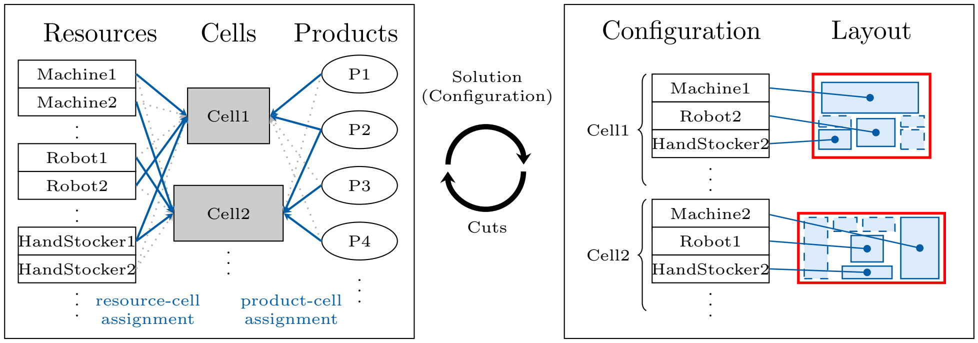
A MILP model for the system configuration problem is proposed as follows:

Minimize

$$\sum_{c,r} K^R_r b_{c,r} + \sum_{c,m,n} K^M_n d_{c,m,n} + \sum_{c,g} K^G_g z^G_{c,g} + \sum_{c,i} K^{GS}_i z^{GS}_{c,i} + \sum_{c,j} K^{JS}_j z^{JS}_{c,j} + \sum_{c,k} K^{PS}_k z^{PS}_{c,k} + \sum_{c,a} K^{AD}_a z^{AD}_{c,a} + \sum_{p,c,r,m,n} y_{p,c,r,m,n} \quad (1)$$

subject to

$$x_{p,c,r,m,n} \leq y_{p,c,r,m,n} \qquad \forall\, p,\, c,\, r,\, m,\, n \qquad (2)$$



(a) Master problem: system configuration.

(b) Sub-problem: layout planning.

Figure 1: Conceptual diagram of the Benders decomposition framework.

$$\sum_{c,r,m,n} x_{p,c,r,m,n} = 1 \qquad \forall\, p \tag{3}$$

$$\sum_{r} b_{c,r} \leq 1 \qquad \forall\, c \tag{4}$$

$$\sum_{n} d_{c,m,n} \leq \sum_{r} b_{c,r} \qquad \forall\, c,\, m \tag{5}$$

$$\sum_{n} d_{c,m,n} \leq 1 - b_{c,r} \qquad \forall\, c,\, r,\, m > \overline{M}_r \tag{6}$$

$$y_{p,c,r,m,n} \leq b_{c,r} \qquad \forall\, p,\, c,\, n,\, m,\, r \in J_p \tag{7}$$

$$\sum_{c,m,n} y_{p,c,r,m,n} = 0 \qquad \forall\, p,\, r \notin J_p \tag{8}$$

$$y_{p,c,r,m,n} \leq d_{c,m,n} \qquad \forall\, p,\, c,\, r,\, m,\, n \in N_p \tag{9}$$

$$\sum_{c,r,m} y_{p,c,r,m,n} = 0 \qquad \forall\, p,\, n \notin N_p \tag{10}$$

$$y_{p,c,r,m,n} \leq \sum_{g \in G_p} z^{G}_{c,g} \qquad \forall\, p,\, c,\, r,\, m,\, n \tag{11}$$

$$\sum_{k} z^{PS}_{c,k} \geq \sum_{r} b_{c,r} \qquad \forall\, c \tag{12}$$

$$(q_c = 0) \implies \sum_{g} z^{G}_{c,g} \leq 1 \qquad \forall\, c \tag{13}$$

$$(q_c = 1) \implies \sum_{p,r,m,n} 2D_p T^G_{p,r} x_{p,c,r,m,n} \le t^G_c \qquad \forall\, c \tag{14}$$

$$(q_c = 1) \implies \sum_i z^{GS}_{c,i} \ge \sum_g z^G_{c,g} - 1 \qquad \forall\, c \tag{15}$$

$$\sum_j z^{JS}_{c,j} \ge \sum_r b_{c,r} \qquad \forall\, c \tag{16}$$

$$\sum_a z^{AD}_{c,a} \ge \sum_r b_{c,r} \qquad \forall\, c \tag{17}$$

$$\sum_{p,r,n} (T^J_{p,n,r} y_{p,c,r,m,n} + (T^L_{p,r} + T^P_{p,n} + T^U_{p,r}) D_p x_{p,c,r,m,n}) \le \theta \sum_r b_{c,r} \qquad \forall c, m \tag{18}$$

$$\sum_{p,r,m,n} (T^J_{p,r} y_{p,c,r,m,n} + (T^L_{p,r} + T^U_{p,r}) D_p x_{p,c,r,m,n}) + t^G_c \le \theta \sum_r b_{c,r} \qquad \forall c \tag{19}$$

$$\sum_r S^R_r b_{c,r} + \sum_n S^M_n d_{c,m,n} + \sum_i S^{GS}_i z^{GS}_{c,i} + \sum_j S^{JS}_j z^{JS}_{c,j} + \sum_k S^{PS}_k z^{PS}_{c,k} + \sum_a S^{AD} z^{AD}_{c,a} \le \overline{S}_c \qquad \forall c \tag{20}$$

$$t^G_c \ge 0 \qquad \forall c, h \tag{21}$$

$$y_{p,c,r,m,n}, b_{c,r}, d_{c,m,n}, q_c, z^G_{c,g}, z^{GS}_{c,i}, z^{JS}_{c,j}, z^{PS}_{c,k}, z^{AD}_{c,a} \in \{0,1\} \qquad \forall p,\, c,\, r,\, m,\, n,\, g,\, i,\, j,\, k,\, a \tag{22}$$

The objective is to minimize the sum of investment costs and product-cell assignment penalties (1). The product-cell assignment penalty is included in the objective to ensure the assignment of the same type of product to the same cell as much as possible. If a fraction of the demand for product p is satisfied by machine n in slot m and robot r in cell c, then product p is assigned to the same machine, slot, robot and cell (2). All demand is satisfied without any stock or backlog (3). At most one robot is assigned to a cell (4), and if a robot is assigned to a cell, the cell is considered built, and machines can be assigned to the built cell (5). There is an upper limit on the number of machines allocated to a cell, determined by the type of the assigned robot (6). A product can be assigned to a cell with the appropriate robot (7,8), machine (9,10) and gripper (11). Each cell has a part stocker to store parts before and after processing them (12). No gripper change time is calculated for single gripper cells (13), whereas multi-gripper cells require gripper change time (14) and one less gripper stocker than the number of assigned grippers (15). Each built cell requires a jig stocker (16) and an adjustment device (17). The machine load, consisting of jig change time, part loading and unloading times, and process time, must not exceed the duration of the given production period (18). The robot load, which consists of jig change time, part loading and unloading times, and gripper change time, must not exceed the given production period (19). The total floor space required by the resources allocated to a cell must not exceed the available floor space of the cell (20). Binary variables are enumerated in constraint (22).

4.2. MILP model for layout planning problem

The MILP model for the layout planning problem extends the MILP of [13] by minimizing the robot travel distance. The MILP is formulated as follows:

$$\text{minimize } d^L + d^U + d^J + d^G \tag{23}$$

subject to

$$w_i = (1 - z_i)W_i + z_i H_i \qquad \forall\, i \in \mathcal{I}_c \tag{24}$$

$$h_i = (1 - z_i)H_i + z_i W_i \qquad \forall\, i \in \mathcal{I}_c \tag{25}$$

$$x_i \leq \frac{w_i}{2} \qquad \forall\, i \in \mathcal{I}_c \tag{26}$$

$$x_i \leq \overline{W}_c - \frac{w_i}{2} \qquad \forall\, i \in \mathcal{I}_c \tag{27}$$

$$y_i \geq \frac{h_i}{2} \qquad \forall\, i \in \mathcal{I}_c \tag{28}$$

$$y_i \leq \overline{H}_c - \frac{h_i}{2} \qquad \forall\, i \in \mathcal{I}_c \tag{29}$$

$$\alpha_{i,j} \leq 1 - \alpha_{j,i} \qquad \forall i, j \in \mathcal{I}_c\colon i \neq j \tag{30}$$

$$\rho_{i,j} \leq 1 - \rho_{j,i} \qquad \forall i, j \in \mathcal{I}_c\colon i \neq j \tag{31}$$

$$\alpha_{i,j} + \rho_{i,j} \geq 1 - \alpha_{j,i} - \rho_{j,i} \qquad \forall i, j \in \mathcal{I}_c\colon i \neq j \tag{32}$$

$$(\alpha_{i,j} = 1) \implies y_i \geq y_j + \frac{h_i + h_j}{2} + M_{i,j} \qquad \forall i, j \in \mathcal{I}_c\colon i \neq j \tag{33}$$

$$(\rho_{i,j} = 1) \implies x_i \ \geq \ x_j + \frac{w_i + w_j}{2} + M_{i,j} \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (34)$$

$$d_{i,j} = \delta^x_{ij} + \delta^y_{ij} \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (35)$$

$$\delta^x_{i,j} \geq x_i - x_j \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (36)$$

$$\delta^x_{i,j} \geq x_j - x_i \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (37)$$

$$\delta^y_{i,j} \geq y_i - y_j \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (38)$$

$$\delta^y_{i,j} \geq y_j - y_i \qquad \forall i, j \in \mathcal{I}_c \colon i \neq j \quad (39)$$

$$d^L = \sum_{\substack{i \in \mathcal{R}_c, \\ j \in \mathcal{PS}_c}} d_{i,j} + \sum_{\substack{i \in \mathcal{PS}_c, \\ j \in \mathcal{AD}_c}} d_{i,j} + \sum_{\substack{i \in \mathcal{AD}_c, \\ j \in \mathcal{M}_c}} d_{i,j} + \sum_{\substack{i \in \mathcal{M}_c, \\ j \in \mathcal{R}_c}} d_{i,j} \quad (40)$$

$$d^U = \sum_{\substack{i \in \mathcal{R}_c, \\ j \in \mathcal{M}_c}} d_{i,j} + \sum_{\substack{i \in \mathcal{M}_c, \\ j \in \mathcal{PS}_c}} d_{i,j} + \sum_{\substack{i \in \mathcal{PS}_c, \\ j \in \mathcal{R}_c}} d_{i,j} \quad (41)$$

$$d^J = 2 \sum_{\substack{i \in \mathcal{R}_c, \\ j \in \mathcal{M}_c}} d_{i,j} + 2 \sum_{\substack{i \in \mathcal{M}_c, \\ j \in \mathcal{JS}_c}} d_{i,j} \quad (42)$$

$$d^G = 2 \sum_{\substack{i \in \mathcal{R}_c, \\ j \in \mathcal{GS}_c}} d_{i,j} \quad (43)$$

$$\alpha_{i,j}, \rho_{i,j} \ \in \{0, 1\} \qquad \forall i, \ j \in \mathcal{I}_c \colon i \neq j \quad (44)$$

$$z_i \ \in \{0, 1\} \qquad \forall \, i \in \mathcal{I}_c \quad (45)$$

In order to improve handling efficiency, the objective function (23) of the layout planning problem is to minimize the robot travel distance of the robot handling tasks such as part loading and unloading, jig change, and gripper change. Constraints (24) and (25) define the effective dimensions of items after accounting for rotations. Constraints (26) to (29) indicate that the x and y coordinates of each item are restricted to fit within the cell boundaries. Constraints (30) to (32) ensure that two distinct items are either above, below, on the left or on the right of each other. These relative positions are translated to absolute positions to guarantee that the items do not overlap, including a margin, by Constraints (33) and (34).

The Manhattan-distance between distinct items are determined by Constraints (35) to (39). The path for part loading operation is defined as the sum of the distances from the robot to the part stocker, from the part stocker to the adjustment device, from the adjustment device to the machine, and from the machine to the robot (40). The path for part unloading operation is defined as the sum of the distances from the robot to the machine, from the machine to the part stocker, and from the part stocker to the robot (41). The path for jig change operation is defined as the sum of the distances from the robot to the machine, from the machine to the jig stocker, from the jig stocker to the machine, and from the machine to the robot (42). The path for gripper change operation is defined as the sum of the distances from the robot to the gripper stocker, and from the gripper stocker to the robot (43). Variables representing the rotational state or positional relationship of items are defined as binary values in constraints (44) to (45).

4.3. Benders cuts formulation

The layout planning MILP model outputs either a feasible layout or a set of items that could not be placed together in a cell without overlapping. If the layout planning problem is infeasible for a cell, additional constraints (Benders cuts) are generated and fed back to the system configuration problem. The Benders cuts defined in this section are the ones proposed in [13], adapted to the system configuration problem of 4.1. For the sake of self-containedness, the cuts are described in this section.

Suppose that after solving the system configuration problem and retrieving the optimal solution, the layout planning problem does not find a feasible layout for cell c. Let $\mathcal{D}_c$ denote the set of machine-slot pairs (i.e., machine $m \in \mathcal{M}_c$ and its respective slot is an element of $\mathcal{D}_c$) and $\mathcal{R}_c, \mathcal{GS}_c, \mathcal{PS}_c, \mathcal{AD}_c, \mathcal{JS}_c$ denote the robots, gripper stockers, part stockers, adjustment devices and jig stockers assigned to cell c, respectively and let N denote their total number, i.e., $N = |\mathcal{M}_c| + |\mathcal{R}_c| + |\mathcal{GS}_c| + |\mathcal{PS}_c| + |\mathcal{AD}_c| + |\mathcal{JS}_c|$. Since the cell does not admit a feasible layout, the assigned items cannot be placed simultaneously in the cell. Equivalently, at most $N-1$ of the items can be assigned to the cell in a feasible solution, hence,

$$\sum_{(m,n)\in\mathcal{D}_c} d_{c,m,n} + \sum_{r\in\mathcal{R}_c} b_{c,r} + \sum_{g\in\mathcal{GS}_c} z^{GS}_{c,g} + \sum_{k\in\mathcal{PS}_c} z^{PS}_{c,k} + \sum_{a\in\mathcal{AD}_c} z^{AD}_{c,a} + \sum_{j\in\mathcal{JS}_c} z^{JS}_{c,j} \leq N-1 \tag{46}$$

is a valid inequality for the system configuration problem, and it cuts the current configuration by not letting all the chosen items be assigned to cell c simultaneously. This type of cut is referred to as a *no-good cut*.

If the items assigned to a cell have no feasible layout, replacing one item with a larger item cannot result in a feasible layout either. Therefore, a stronger cut can be derived by extending the item set of the no-good cut with further items that are larger than the original ones. For this purpose, a partial ordering $\succeq$ of the items is defined for items i and j as follows: $i \succeq j$ if and only if $\max\{W_i, H_i\} \geq \max\{W_j, H_j\}$ and $\min\{W_i, H_i\} \geq \min\{W_j, H_j\}$. Then, the inequality

$$\sum_{(m,n)\in\mathcal{D}_c}\sum_{n'\succeq n} d_{c,m,n'} + \sum_{r\in\mathcal{R}_c}\sum_{r'\succeq r} b_{c,r'} + \sum_{g\in\mathcal{GS}_c}\sum_{g'\succeq g} z^{GS}_{c,g'} + \sum_{k\in\mathcal{PS}_c}\sum_{k'\succeq k} z^{PS}_{c,k'} + \sum_{a\in\mathcal{AD}_c}\sum_{a'\succeq a} z^{AD}_{c,a'} + \sum_{j\in\mathcal{JS}_c}\sum_{j'\succeq j} z^{JS}_{c,j'} \leq N-1 \qquad (47)$$

is also valid, it is not satisfied by the current solution, and it dominates the original no-good cut.

Moreover, if a set of items does not fit into cell c, then they do not fit into a smaller cell either. To formalize this observation, a partial ordering similar to the one above is defined over cells by letting $c \succeq c'$ for two cells c and c' if and only if $\max\left\{\overline{W}_c, \overline{H}_c\right\} \geq \max\left\{\overline{W}_{c'}, \overline{H}_{c'}\right\}$ and $\min\left\{\overline{W}_c, \overline{H}_c\right\} \geq \min\left\{\overline{W}_{c'}, \overline{H}_{c'}\right\}$.

By combining the two ideas, i.e., that items cannot be replaced by larger items, and the cell cannot be replaced by a smaller cell, the following, even stronger cut can be derived:

$$\sum_{(m,n)\in\mathcal{D}_c}\sum_{n'\succeq n} d_{c',m,n'} + \sum_{r\in\mathcal{R}_c}\sum_{r'\succeq r} b_{c',r'} + \sum_{g\in\mathcal{GS}_c}\sum_{g'\succeq g} z^{GS}_{c',g'} + \sum_{k\in\mathcal{PS}_c}\sum_{k'\succeq k} z^{PS}_{c',k'} + \sum_{a\in\mathcal{AD}_c}\sum_{a'\succeq a} z^{AD}_{c',a'} + \sum_{j\in\mathcal{JS}_c}\sum_{j'\succeq j} z^{JS}_{c',j'} \leq N-1 \quad \forall c'\colon c' \preceq c. \qquad (48)$$

The above cuts (48) are referred to as *lifted cuts*. The cuts are generated for all cells with no feasible layout, added to the MILP formulation of the system configuration problem, which is then solved again, as depicted in Figure 1. The procedure stops when all cells admit a feasible layout.

5. Computational experiments

The effectiveness of the proposed method was evaluated in numerical experiments involving the design of an automated machining cell system. In these experiments, the following two methods were compared:

(i) The baseline method, which implemented only the no-good cuts (46).

(ii) The advanced method, which implemented the lifted cuts (48).

The details of the problem instances and the experimental conditions are described in Section 5.1, and the experimental results are discussed from the perspective of computational efficiency and industrial effectiveness in Section 5.2.

5.1. Experimental conditions

The resource candidates included 5 types of machines, robots and grippers, 3 types of part stockers, gripper stockers and jig stockers. The 5 types of products have different shapes and sizes, thus, a robot requires a product-specific gripper to grasp the product precisely. The sizes of the gripper stocker and jig stocker were assumed to be such that a higher price allowed more layers to be stacked using vertical space, and the smaller the occupied floor space becomes. The dimensions of the gripper are irrelevant because the gripper itself is placed either on a gripper stocker or attached to a robot. The margin between resources was uniformly set to 200 mm. Ten cells with varying available floor space were defined.

The MILP models and Benders cuts were implemented in Mosel language and FICO Xpress [50] version 9.2 was used to solve the MILP problem. The experiments were performed on a computer with i7-10700K CPU @ 3.80GHz, 64.0 GB RAM, and Windows operating system. The limit of calculation time was set to 3600 seconds for each problem instance.

In order to evaluate the computational efficiency of the Benders method, problem instances were generated by controlling problem size and applying random perturbations to the original data set described in Section 5.1. The problem size was controlled by varying the number of cells $|C| \in \{5, 10\}$ and the number of product types $|P| \in \{5, 10, 20\}$. For each problem size, five random instances were generated, resulting in a total of 30 instances. Specifically, the random perturbations were applied to the production demand volume, process time, loading time, unloading time, and jig change time.

5.2. Results

5.2.1. Computational efficiency

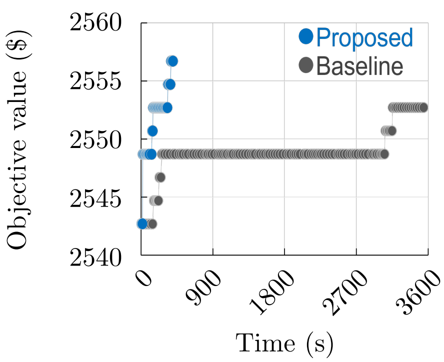
Table 3 shows the computational results averaged on the instances of the different problem sizes. Columns C and P show the number of product types and available cells, respectively. The columns of the table are divided into two groups: (i) "Baseline method" and (ii) "Advanced method". Column "Opt." contains the number of instances for which a solution with proven optimality was found, column "Agv. time ± SD" shows the average calculation time and its standard deviation (SD) in seconds. The average was taken over all instances of a problem size, including those with no optimal solution found within the time limit. The relative difference of the average runtimes of the two methods as percentage is also shown in column "Δtime".

For $C = 5$, on each problem size, the advanced method dominated the baseline method in terms of number of optimal solutions and average running times. For $C = 10$, the advance method found optimal solution 7 times out of 15, while the baseline method only twice. Only the largest problem size proved to be too difficult for the advanced method consistently. Overall, the baseline method found 14 optimal solution, whereas the advance method found 22. That, together with the significant reduction in computational time confirms the effectiveness of the advanced method.

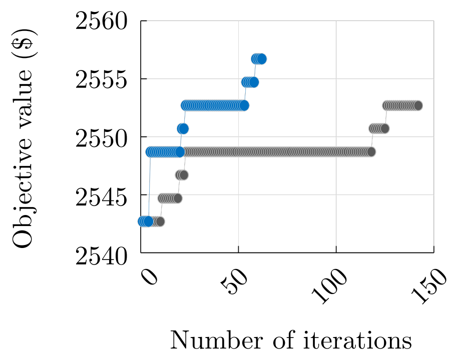
Table 3: Comparison of computation results between baseline and proposed methods.

Parameter		Baseline method			Advanced method			
C	P	Opt	Avg. time ±	SD (s)	Opt	Avg. time ±	SD (s)	Δtime %
5	5	5	527 ±	199	5	145 ±	69	-73%
	10	5	954 ±	262	5	455 ±	126	-52%
	20	2	2,692 ±	1,141	5	1,410 ±	945	-48%
10	5	2	2,807 ±	997	4	1,021 ±	1,313	-64%
	10	0	3,600 ±	0	3	2,965 ±	662	-18%
	20	0	3,600 ±	0	0	3,600 ±	0	0%

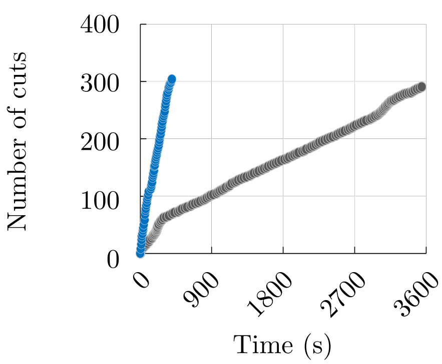
To more clearly demonstrate the effectiveness of the advanced method, detailed results are provided for an instance of size $C = 10$ and $P = 5$, where advanced method achieved the greatest reduction in the absolute value of the average calculation time from 2807 to 1021 seconds. Figures 2a and 2b show the objective value for the two methods compared to the computational time and the number of iterations, respectively. The baseline method was unable to find a solution within one hour, while the advanced method managed to find the optimal solution in 408 seconds. Figures 2c and 2d show the changes in the number of generated cuts compared to the the calculation time and the number of iterations, respectively. In the case of the baseline method, after one hour had elapsed, the number of iterations was 142 and the total number of cuts was 293. On the other hand, in the advanced method, when the optimal solution was found, the number of iterations was 62 and the total number of cuts was 304. These results show that the advanced method generates more cuts per iteration than the baseline method, and that the computational time for solving higher-level problems is reduced due to the effects of these cuts.



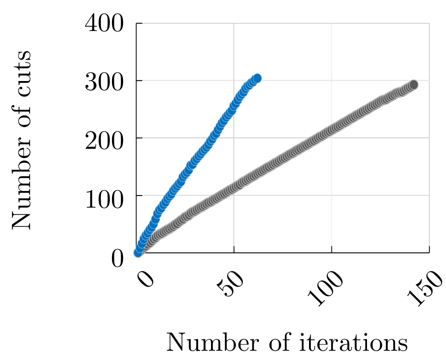
(a) Objective value and computation time.



(b) Objective value and number of iterations.



(c) Total number of generated cuts and computation time.



(d) Total number of generated cuts and number of iterations.

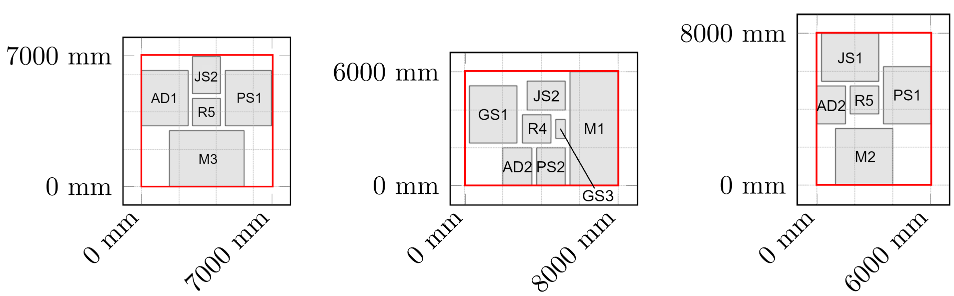
Figure 2: Comparison of baseline and advanced methods for an instance: objective value and total number of cuts versus computation time and number of iterations.

5.2.2. System configuration and layout for an instance

To evaluate the practicality of the system configuration obtained by the proposed method, the detailed configuration of the optimal solution is shown in Figure 3. The physical location of the cells is ignored, and the cells are simply arranged from top left to bottom right. The area enclosed by the red frame represents the available floor space of each cell.

The solution of the system configuration problem proposed building three cells of the available 10 cell locations, Cells 4, 5, and 7. Each cell was appropriately equipped with machine tools, robots, grippers, part stockers, adjustment devices, and jig stockers. In Cell 5, multiple grippers are assigned to the robot, hence, gripper stockers are appropriately assigned to store unused grippers. The obtained results match the acceptable configuration that can be used in actual manufacturing sites, and the practicality of the solution method was confirmed.

In Cell 4, five items are arranged within the available floor space of the cell. Robot R5 placed in the center, transporting parts to machine M3, part stocker PS1 and adjustment device AD1, also transporting and placing jigs in jig stocker JS2. Since Cell 4 is a single-gripper cell, no gripper stocker is needed as the robot is always equipped with the only gripper. There is a margin between resources as space for maintenance work by human in case a resource fails or stops during daily operation. As well as Cell 4, the resources are arranged around the robot and arranged within the available floor space in Cell 5. A difference from Cell 4 is that two gripper stockers (GS1 and GS3) have been assigned to the cell to store the gripper not attached to the robot.



(a) Layout of Cell 4. (b) Layout of Cell 5. (c) Layout of Cell 7.

Figure 3: Visualization of layout for an instance.

6. Industrial case study

6.1. Development of a line design software tool

After confirming the computational efficiency of the proposed methods, a line design software tool incorporating these methods was implemented, see Figures 4 and 5. In the developed workflow, the engineer first manually prepares the input data in Excel tables and uploads them to the web application, which stores them in a relational database. The engineer can define multiple scenarios to explore possible configurations and layouts (Figure 4). For this purpose, the web application invokes an implementation of the proposed MILP models and algorithms using FICO Xpress. When the calculation completes—either by reaching an optimal solution or by hitting the time limit—the engineer reviews the solver output in the web application, including the 2D layout diagram as shown in Figure 5.

If the solution satisfies all requirements of the engineer, the system configuration and layout information is exported as an XML file, serving as the interface to the commercial 3D simulator Visual Components [51]. If the solution requires further adjustments, these can be performed either in the web application by adjusting the input parameters of the solver, or directly in the 3D simulator. This iterative workflow is presented later in Section 6.3.2.

Figure 4: Web-based graphical user interface of the developed decision support tool: definition of multiple scenarios for what-if-analysis.

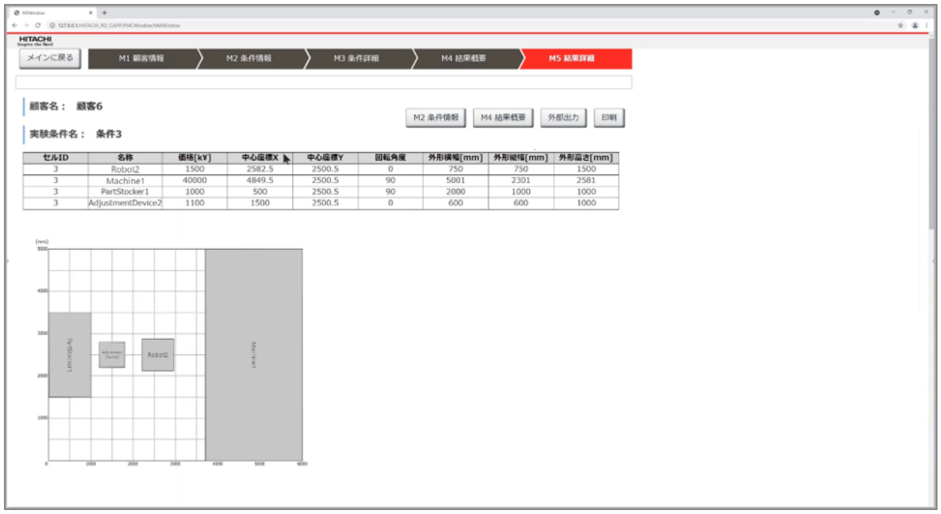


Figure 5: Web-based graphical user interface of the developed decision support tool: 2D visualization of the configuration and layout.

6.2. Problem instance

The sample problem instance includes a single product with a demand of 7200 units per year. The process time for each machine type was calculated based on CAM and skilled workers' input, resulting in values around 30 minutes per unit. The higher the machine cost, the shorter the process time tended to be. As shown in Table 4, the resource candidates included 5 types of machines, 5 types of robots, 5 types of grippers, 3 types of part stockers, and 3 types of jig stockers. Since only one type of product was involved, a single gripper sufficed, eliminating the need for a gripper stocker. The higher the price of a part stocker, adjustment device, and jig stocker, the more layers can be stacked using the vertical space, and the smaller floor space they occupy. The dimensions of the gripper are not defined because the gripper itself is attached to the robot. The instance included a single available cell with a rectangular floor space measuring 7,000 mm by 5,000 mm. The minimum margin between resources was set to 600 mm. Note that these item properties are different from those used in the computational experiments.

Table 4: Resource candidates in the industrial case study.

Category	Name	Purchase price ($)	Width (mm)	Height (mm)
Machine	M1	573,000	4,410	2,700
	M2	498,000	5,221	4,811
	M3	673,000	5,425	2,995
	M4	434,000	2,970	3,831
	M5	348,000	3,865	1,842
Robot	R1	10,000	1,500	1,500
	R2	15,000	1,500	1,500
	R3	20,000	1,500	1,500
	R4	25,000	1,500	1,500
	R5	30,000	1,500	1,500
Gripper Hand	G1	5,000	-	-
	G2	6,000	-	-
	G3	7,000	-	-
	G4	9,000	-	-
	G5	10,000	-	-
Part stocker	PS1	14,000	2,300	900
	PS2	12,000	2,500	1,000
	PS3	10,000	3,000	1,500
Adjustment device	AD1	9,000	700	700
	AD2	6,000	1,400	1,400
	AD3	4,000	2,000	2,000
Jig stocker	JS1	14,000	700	700
	JS2	10,000	1,400	700
	JS3	6,000	1,400	1,400

6.3. Results

6.3.1. System configuration and layout

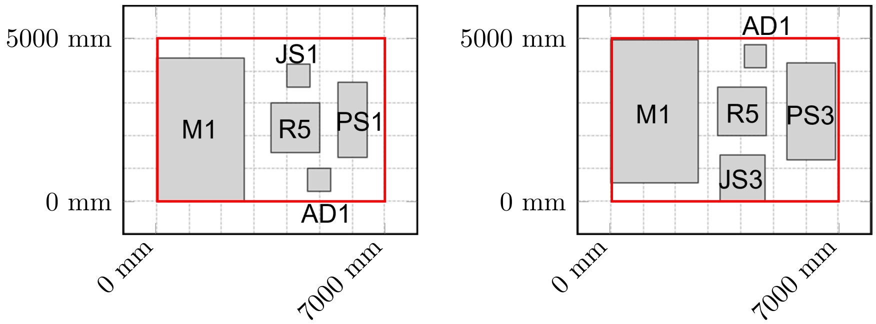
The results of conventional manual design are shown in Table 5 and Figure 6a, whereas those of the proposed automated method are presented in Table 6 and Figure 6b. Both solutions satisfy all technological and geometric constraints of the model. Yet, the solutions apply slightly different configurations: the automated solution included a cheaper but larger part stocker (PS3 vs. PS1) and jig stocker (JS3 vs. JS1) compared to the manual design. This configuration still satisfies floor space constraints while it incurs a 1.7% lower investment cost (634,000$ vs. 646,000$). Despite the use of larger items, the automated solution also decreased the robot travel distance by 1.8%, from 38,070 mm to 37,400 mm.

Table 5: Results obtained through manual design. Columns x and y indicate the position of the midpoints the items.

Resource name	Investment cost ($)	Width (mm)	Height (mm)	x (mm)	y (mm)	Rotated
M1	573,000	4,410	2,700	1,350	2,205	90°
R5	30,000	1,500	1,500	4,250	2,250	0°
G2	6,000	-	-	-	-	-
PS1	14,000	2,300	900	6,000	2,500	90°
AD1	9,000	700	700	5,000	650	0°
JS1	14,000	700	700	4,350	3,850	0°

Table 6: Results obtained by the proposed automated methods. Columns x and y indicate the position of the midpoints of the items.

Resource name	Investment cost ($)	Width (mm)	Height (mm)	x (mm)	y (mm)	Rotated
M1	573,000	4,410	2,700	1,350	2,795	90°
R5	30,000	1,500	1,500	4,050	2,795	0°
G2	6,000	-	-	-	-	-
PS3	10,000	3,000	1,500	6,150	2,795	90°
AD1	9,000	700	700	4,450	4,495	0°
JS3	6,000	1,400	1,400	4,000	745	0°



(a) Manual design. (b) Automated design.

Figure 6: Solutions obtained via manual design (a) and by the proposed automated method (b).

6.3.2. Manual adjustments

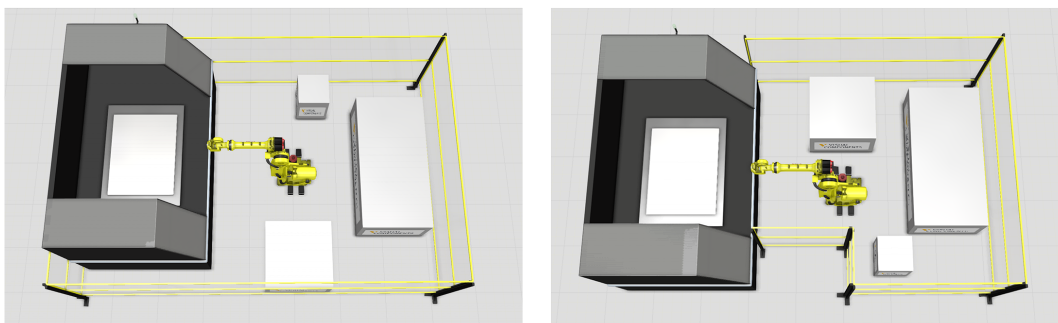
Figure 7 shows the 3D layouts before and after the manual modifications by the engineer. The layout before modifications (Figure 7a) was automatically generated by the solver. Based on their background knowledge, the engineer made the following modifications:

- In the layout before modification, safety fences were installed around the robot to prevent collisions with human operators (Figure 7a). The engineer repositioned the posts of the fence to reduce the total length, which resulted in a slight reduction in investment cost (Figure 7b).

- In general, machines equipped with automatic tool-changing functions have tool magazines arranged on their sides to store cutting tools, such as drills, milling cutters, and turning tools. The selected machine has a tool magazine, which is not depicted clearly in the model in order to protect confidential information. Access to the tool magazine for maintenance and cleaning required the operator to stop the robot in the original layout (Figure 7a). In order to allow the operator to access the tool magazine from outside the safety fence, the engineers repositioned the fence in the area below the robot (Figure 7b). This repositioning was not essential, but the engineers decided that it would satisfy the floor space constraints and improve accessibility to the tool storage magazine.

- The space in front of the machine tool magazine was reduced by repositioning the safety fence, so the positions of the jig stocker and the smaller adjustment device were swapped.

- Finally, the layout was fine-tuned by moving items as close to the robot as possible while ensuring that the movement of the robot is not hindered and that all items are accessible by the human operator for maintenance. It is noted that while the resulting distances slightly violate the minimum margins defined in the input of the solver, the engineer judged that the overall layout still satisfies all accessibility requirements.



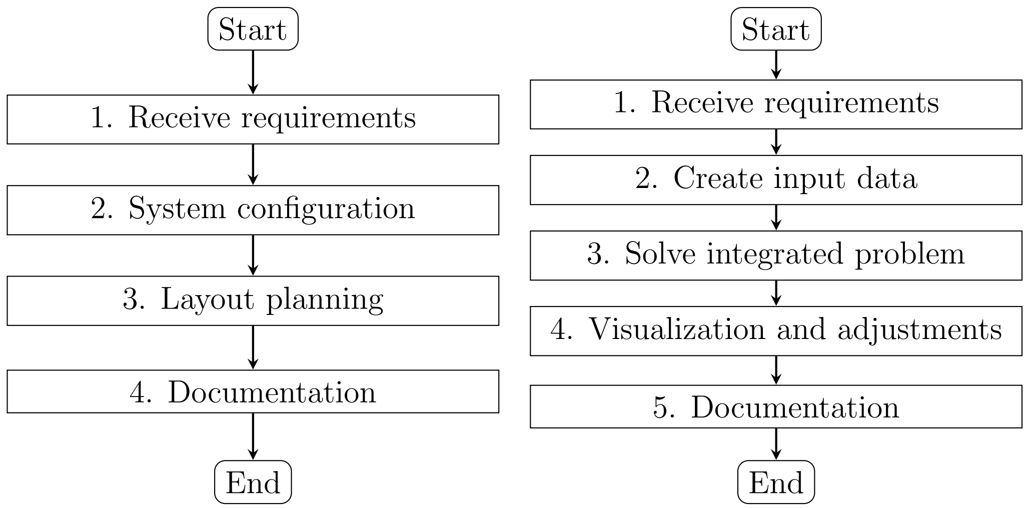
(a) Layout from solver. (b) Manually modified layout.

Figure 7: Comparison between the layout computed by the solver (a) and the layout adjusted manually (b).

6.3.3. Comparison of workflow and man-hours

As shown in Figure 8a, the conventional workflow of line design contains the following steps:

1. Receive requirements from customer.
2. Design the system configuration. This process takes about 10 hours, and the system configuration is developed based on the engineer's experience and past cases.
3. Design the layout based on the system configuration. The layout is designed over about 8 hours, while revising the system configuration as necessary.



(a) As-is manual design workflow. (b) Proposed design workflow.

Figure 8: Comparison between the as-is manual design workflow (a) and the proposed design workflow using the developed tool (b).

4. Create documents to present to customer. The engineer estimated that it takes about 4 hours to create documents to explain the results to the customer.

In total, this process required about 22 hours of work.

On the other hand, as shown in Figure 8b, the workflow using the newly introduced line design support tool is as follows:

1. Receive requirements from customer.

2. Create input data. The developed tool supports the engineer in preparing the input data, which hence takes only 7 minutes.

3. Solve the integrated problem. Finding the optimal configuration and layout using the proposed solution approach takes 1 minute.

4. Visualization and adjustments. The results from the automated solver are automatically converted into a 3D simulation model in Visual Components, where engineers make all necessary adjustments. This process took 19 minutes.

5. Create documents to present to the customer. As before, this was estimated to take 4 hours.

Hence, the newly introduced workflow reduced the total man-hours for system design to 4.5 hours, compared to the 22 hours required using the conventional manual workflow, see Figure 9. This significant improvement in efficiency is due to the simultaneous optimization of system configuration and layout, as well as the rapid layout confirmation in the 3D simulator. This has freed up engineers from repetitive tasks, allowing them to respond to customers with system design results in a shorter amount of time.

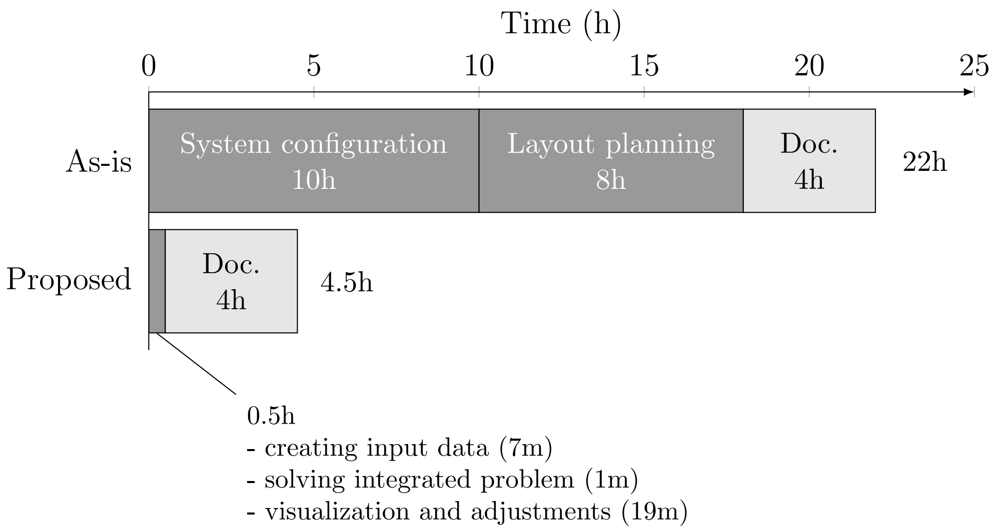


Figure 9: Comparison of the man-hours required for system design using the as-is manual workflow and the proposed workflow using the developed tool.

6.4. Sensitivity analysis

Depending on the requirements of different customers, the values of input parameters for the proposed model may naturally differ from the values defined in the case study above. Therefore, a sensitivity analysis was performed in order to evaluate the effect of input parameter values on solutions, as well as to provide further insight into model robustness.

6.4.1. Experimental setup

The instance of the industrial case study described in Section 6.2 was selected as the baseline. Three key groups of parameters were selected for variation in the experiments as follows:

(i) Process times $T^P_{p,n}$ on the machines. A significant increase in process times implies that either faster and more expensive machines, or a higher number of machines is required to satisfy the same demand volume.

(ii) Handling times by the robots, including part loading times $T^L_{p,r}$, part unloading times $T^U_{p,r}$, gripper change times $T^G_{p,r}$, and jig change times $T^J_{p,n,r}$. A significant increase in handling times implies that a more powerful, and accordingly, more expensive robot is required to satisfy the given demand.

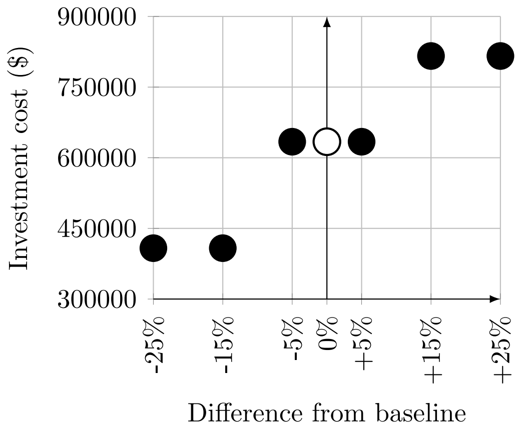
(iii) Cell sizes. For example, a 5% increase in cell size means that the width and height of a cell are simultaneously increased by 5%, resulting in the same aspect ratio, and a floor space $\overline{S}_c$ 10.25% larger than the original value.

For each parameter group, six additional experimental runs were performed with ±5%, ±15%, and ±25% deviation from the baseline value.

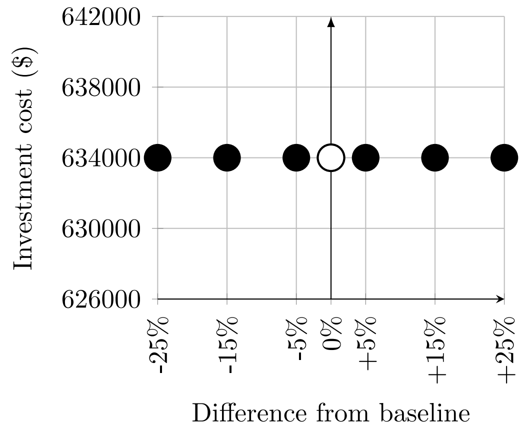
6.4.2. Results and discussion

The results of the sensitivity analysis are shown in Figure 10. The figure is composed of three graphs corresponding to the key parameter groups: (a) process times, (b) handling times, and (c) cell sizes. In each graph, the horizontal axis indicates the difference of the parameter values from the baseline in percents, and the vertical axis shows the resulting investment cost. Each graph uses a unique vertical-axis scale to enhance the readability of the results.

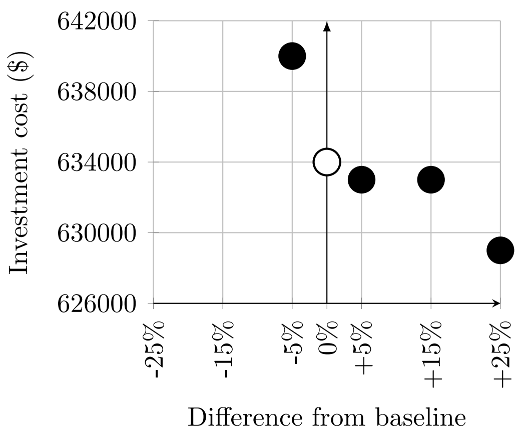
Figure 10a shows that variations in process times caused step-like changes in investment costs. Small changes (±5%) had no impact on the cost, whereas larger changes (±15% and ±25%) significantly altered it. The increase from 0% to +15% raised the cost from 634,000$ to 816,000$, since one additional cell was required to satisfy the demand volume. However, the cost did not simply double from the baseline, because the machine M1 of the baseline was replaced with the cheaper machine M5. The increase from +15% to +25%



(a) Sensitivity to variations in process times.



(b) Sensitivity to variations in handling times.



(c) Sensitivity to variations in cell size.

Figure 10: Results of sensitivity analysis. Solid marks indicate the perturbed instances, empty mark indicates the baseline. Optimal solutions were found for all conditions except for –15% and –25% cell size.

did not alter the cost; however, a further increase in process times would gradually require more and more powerful machines and additional cells at the price of higher investment costs.

A –15% reduction of process times decreased the cost to 408,000$ in the single cell, mainly due to the replacement of machine M1 with the cheaper alternative M5, which was sufficient to satisfy the given demand in the modified scenario. In parallel, adjustment device AD1 was replaced with AD3 (significantly cheaper but larger than AD1), whereas fixture stocker FS3 was replaced by FS2 (somewhat more expensive but smaller than FS3). Overall, the application of the new, less expensive combination of AD3 and FS2 was made possible by using the smaller machine M5. A further decrease to –25% had no impact on the cost because the solution at –15% already applied the lowest-cost machine M5.

Figure 10b illustrates that variation in robot handling times did not influence investment costs. Even large changes (±15% and ±25%) had no impact, because handling times (e.g., part loading and unloading times of approximately 2 minutes) were far from being a bottleneck. It should be noted that in other applications, e.g., where a single robot serves many machines in large cells, the model may become sensitive to handling times as well.

Figure 10c presents an inverse relationship between cell size and investment cost. Even small changes (±5% in width and height, corresponding to ±10.25% in floor space) altered the optimal solution, indicating that the model is rather sensitive to cell size. An increase of +5% reduced the cost to 633,000$ by replacing the adjustment device AD1 with the larger but cheaper AD2. A further increase to +25% decreased the cost to 629,000$ by replacing part stocker PS2 and adjustment device AD2 with PS3 and AD3, respectively. Since the solution at +25% already contains the cheapest resources, any further increase in cell size will have no impact on the solution.

The decrease in cell size to –5% raised the investment cost to 640,000$ because the reduction in floor space necessitated the selection of smaller but more expensive resources. Accordingly, part stocker PS3 and fixture stocker FS3 were replaced by PS2 and FS2, respectively. A further decrease to –15% or –25% rendered the problem infeasible due to insufficient floor space.

In summary, the results of the sensitivity analysis show that the model, at least in the current application, is robust to variations in handling times. At the same time, it is sensitive to process times and cell size, since these two factors are the bottlenecks in production system configuration. Even small changes (±5%) in cell size alter the optimal solution, and a larger decrease can render the problem instance infeasible. These findings further underscore the importance of integrated system configuration and layout planning.

7. Conclusions and future work

7.1. Conclusions

This study presented a logic-based Benders decomposition framework for solving the integrated production system configuration and layout planning problem, as well as a detailed case study about the application of the approach in a real industrial environment. The system configuration problem was treated as the Benders master problem, whereas the layout problems related to individual cells as Benders subproblems. Problem-specific lifted cuts ensured the computational efficiency of the approach, which was illustrated in computational experiments on randomized problem instances with up to 20 products and 10 cells.

Even more importantly, an industrial case study was conducted to demonstrate the applicability of the approach in a real-world industrial setting. The integration of the automated planning tool into the overall planning workflow was described in detail, along with the modifications made by the human expert to the automated solution before its physical implementation. The results of the conventional manual planning workflow were also compared to those obtained using the automated planning tool. In the case study, the automated planning tool reduced the man-hours required for system design from 22 hours to 4.5 hours, compared to the conventional manual workflow. These results confirm that the proposed approach is an effective tool for designing cellular manufacturing systems that minimize investment cost, considering all constraints that stem from required throughput, technology, and limited floor space.

7.2. Future work

We consider the following four directions of particular interest for future research: (A) metaheuristics for solving large instances, (B) process division into multiple cells, (C) integration with scheduling, and (D) multi-objective optimization including additional key performance indicators (KPIs) such as environmental aspects.

While the mathematical programming solution approach presented in this paper could solve instances of industrially relevant size to proven optimality, in applications involving significantly larger production systems, computational efficiency can pose a significant challenge. Accordingly, the development of metaheuristic solution techniques for obtaining close-to-optimal solutions for large instances (A) is an important direction for future research. This is particularly relevant for the configuration problem (Benders master problem), since the layout problem (Benders subproblem) could be solved orders of magnitude faster, and the size of the latter, i.e., the number of items to be placed in a single cell, is not expected to increase significantly. Matheuristics, such as large neighborhood search, can be particularly attractive, since these can be implemented on top of the current MILP model with relatively low effort.

Regarding (B), this research assumed that each product completes all processes within a single cell. In order to generalize the proposed method to applications where a single product is processed across multiple cells, it is necessary to create a layout plan for the entire system that takes into account the transfer time between cells based on the process order. Using the terminology of FLP, this requires the integration of inter-cell and intra-cell layout planning with system configuration.

Next, as for (C), this paper addressed production system configuration separately from the planning problems related to the line operation stage, such as detailed production scheduling. Consequently, certain aspects of the system behavior, such as changeover times, had to be estimated without precise operational-phase data. This can be particularly problematic in case of gripper changeovers, which can occur frequently, e.g., if a single robot moving along a linear rail serves a high number of machines. The key challenge, therefore, is to provide accurate foresight into the operational stage, including possible future production schedules, in order to estimate changeover times more reliably.

Finally, regarding (D), in recent years, there has been a growing need for environmentally friendly products that aim to achieve carbon neutrality, and there is demand from company executives for a production line design that takes into account new KPIs such as energy consumption. One of the issues for the future is to expand the proposed method, which aims to minimize initial investment costs, into a multi-objective optimization approach that also considers operational costs, including energy consumption. For this purpose, it is necessary to take on the challenge of predicting energy consumption even in the early line design stage, before any measurements on the particular physical system could be performed. In applications where a new system is built from known equipment, a possible approach for this is to apply energy state models to characterize the consumption of all relevant equipment, and to integrate the energy state models directly into the mathematical model of production system configuration. Addressing these issues is expected to lead to more practical and sustainable production system designs.

Acknowledgments

Hungarian authors were supported by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy" and by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002).

CRediT authorship contribution statement

Hiroyuki Sakata: methodology, software, investigation, validation, writing–original draft. **Péter Dobrovoczki:** methodology, software, formal analysis, writing–review & editing. **Daisuke Tsutsumi:** supervision, conceptualization, methodology, validation. **András Kovács:** conceptualization, methodology, writing–review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Acemoglu, P. Restrepo, Demographics and automation, The Review of Economic Studies 89 (1) (2021) 1–44. doi:10.1093/restud/rdab031.

[2] R. Dekle, Robots and industrial labor: Evidence from Japan, Journal of the Japanese and International Economies 58 (2020) 101108. doi:10.1016/j.jjie.2020.101108.

[3] E. Dinlersoz, Z. Wolf, Automation, labor share, and productivity: plant-level evidence from U.S. manufacturing, Economics of Innovation and New Technology 33 (4) (2023) 604–626. doi:10.1080/10438599.2023.2233081.

[4] G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, G. Chryssolouris, Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach, CIRP Journal of Manufacturing Science and Technology 2 (2) (2010) 81–91. doi:10.1016/j.cirpj.2009.12.001.

[5] S. Y. Nof (Ed.), Springer Handbook of Automation, Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-78831-7.

[6] L. G. Barajas, S. R. Biller, F. Gu, C. Yuan, Virtual launch & validation of manufacturing automation controls, in: 2010 IEEE International Conference on Automation Science and Engineering, 2010, pp. 412–419. doi:10.1109/COASE.2010.5583977.

[7] A. Salihbegovic, Z. Cico, V. Marinkovi, E. Karavdi, Software engineering approach in the design and development of the industrial automation systems, in: Proceedings of the 2008 International Workshop on Software Engineering in East and South Europe, ICSE '08, ACM, 2008, pp. 15–22. doi:10.1145/1370868.1370872.

[8] Y. Koren, S. J. Hu, T. W. Weber, Impact of manufacturing system configuration on performance, CIRP Annals 47 (1) (1998) 369–372. doi:10.1016/s0007-8506(07)62853-4.

[9] P. Pérez-Gosende, J. Mula, M. Díaz-Madroñero, Facility layout planning. An extended literature review, International Journal of Production Research 59 (12) (2021) 3777–3816. doi:10.1080/00207543.2021.1897176.

[10] O. Battaïa, A. Dolgui, A taxonomy of line balancing problems and their solution approaches, International Journal of Production Economics 142 (2) (2013) 259–277. doi:10.1016/j.ijpe.2012.10.020.

[11] M. Albus, M. F. Huber, Resource reconfiguration and optimization in brownfield constrained robotic assembly line balancing problems, Journal of Manufacturing Systems 67 (2023) 132–142.

[12] J. Hooker, G. Ottosson, Logic-based benders decomposition, Mathematical Programming 96 (1) (2003) 33–60. doi:10.1007/s10107-003-0375-9.

[13] P. Dobrovoczki, A. Kovács, H. Sakata, D. Tsutsumi, Integrated system configuration and layout planning for flexible manufacturing systems, Journal of Manufacturing Systems 77 (2024) 384–397. doi:10.1016/j.jmsy.2024.09.020.

[14] O. Battaïa, X. Delorme, A. Dolgui, G. Frederic, B. Finel, Flow line balancing problem: A survey, in: 2015 International Conference on Industrial Engineering and Systems Management (IESM), Vol. 44, IEEE, 2015, p. 1065–1071. doi:10.1109/iesm.2015.7380287.

[15] F. M. Defersha, M. Chen, A comprehensive mathematical model for the design of cellular manufacturing systems, International Journal of Production Economics 103 (2) (2006) 767–783. doi:10.1016/j.ijpe.2005.10.008.

[16] N. Boysen, P. Schulze, A. Scholl, Assembly line balancing: What happened in the last fifteen years?, European Journal of Operational Research 301 (3) (2022) 797–814. doi:10.1016/j.ejor.2021.11.043.

[17] C. Becker, A. Scholl, A survey on problems and methods in generalized assembly line balancing, European Journal of Operational Research 168 (3) (2006) 694–715. doi:10.1016/j.ejor.2004.07.023.

[18] A. Dolgui, B. Finel, N. N. Guschinsky, G. M. Levin, F. B. Vernadat, MIP approach to balancing transfer lines with blocks of parallel operations, IIE Transactions 38 (10) (2006) 869–882. doi:10.1080/07408170500531334.

[19] G. Choi, A goal programming mixed-model line balancing for processing time and physical workload, Computers & Industrial Engineering 57 (1) (2009) 395–400. doi:10.1016/j.cie.2009.01.001.

[20] H. Kim, S. Park, A strong cutting plane algorithm for the robotic assembly line balancing problem, International Journal of Production Research 33 (8) (1995) 2311–2323. doi:10.1080/00207549508904817.

[21] J. Dou, X. Dai, Z. Meng, Graph theory-based approach to optimize single-product flow-line configurations of RMS, The International Journal of Advanced Manufacturing Technology 41 (9–10) (2008) 916–931. doi:10.1007/s00170-008-1541-2.

[22] A. Baykasoğlu, L. Özbakır, Stochastic U-line balancing using genetic algorithms, The International Journal of Advanced Manufacturing Technology 32 (1–2) (2006) 139–147. doi:10.1007/s00170-005-0322-4.

[23] J. Bautista, J. Pereira, Ant algorithms for a time and space constrained assembly line balancing problem, European Journal of Operational Research 177 (3) (2007) 2016–2032. doi:10.1016/j.ejor.2005.12.017.

[24] A. C. Nearchou, Maximizing production rate and workload smoothing in assembly lines using particle swarm optimization, International Journal of Production Economics 129 (2) (2011) 242–250. doi:10.1016/j.ijpe.2010.10.016.

[25] D. Tsutsumi, A. Kovács, Á. Szalóki, Novel heuristic approach to integrating task sequencing and production system configuration, Procedia CIRP 107 (2022) 28–33. doi:10.1016/j.procir.2022.04.005.

[26] G. Papaioannou, J. M. Wilson, The evolution of cell formation problem methodologies based on recent studies (1997–2008): Review and directions for future research, European Journal of Operational Research 206 (3) (2010) 509–521. doi:10.1016/j.ejor.2009.10.020.

[27] T. Aljuneidi, A. A. Bulgak, Designing a cellular manufacturing system featuring remanufacturing, recycling, and disposal options: A mathematical modeling approach, CIRP Journal of Manufacturing Science and Technology 19 (2017) 25–35. doi:10.1016/j.cirpj.2017.04.005.

[28] D. Cao, M. Chen, A robust cell formation approach for varying product demands, International Journal of Production Research 43 (8) (2005) 1587–1605. doi:10.1080/00207540412331327754.

[29] E. Mehdizadeh, S. V. Daei Niaki, V. Rahimi, A vibration damping optimization algorithm for solving a new multi-objective dynamic cell formation problem with workers training, Computers & Industrial Engineering 101 (2016) 35–52. doi:10.1016/j.cie.2016.08.012.

[30] S. M. Saad, A. Baykasoglu, N. N. Z. Gindy, An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells, Production Planning & Control 13 (4) (2002) 381–393. doi:10.1080/09537280210130478.

[31] P. Burggräf, T. Adlon, V. Hahn, T. Schulz-Isenbeck, Fields of action towards automated facility layout design and optimization in factory planning – a systematic literature review, CIRP Journal of Manufacturing Science and Technology 35 (2021) 864–871. doi:10.1016/j.cirpj.2021.09.013.

[32] H. Hosseini-Nasab, S. Fereidouni, S. M. T. Fatemi Ghomi, M. B. Fakhrzad, Classification of facility layout problems: a review study, The International Journal of Advanced Manufacturing Technology 94 (1–4) (2017) 957–977. doi:10.1007/s00170-017-0895-8.

[33] M. Braglia, S. Zanoni, L. Zavanella, Layout design in dynamic environments: analytical issues, International Transactions in Operational Research 12 (1) (2005) 1–19. doi:10.1111/j.1475-3995.2005.00487.x.

[34] A. Tayal, S. P. Singh, Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem, Annals of Operations Research 270 (1–2) (2016) 489–514. doi:10.1007/s10479-016-2237-x.

[35] S. Vitayasak, P. Pongcharoen, C. Hicks, Robust machine layout design under dynamic environment: Dynamic customer demand and machine maintenance, Expert Systems with Applications: X 3 (2019) 100015. doi:10.1016/j.eswax.2019.100015.

[36] J. Feng, A. Che, Novel integer linear programming models for the facility layout problem with fixed-size rectangular departments, Computers & Operations Research 95 (2018) 163–171. doi:10.1016/j.cor.2018.03.013.

[37] S. Kang, J. Chae, Harmony search for the layout design of an unequal area facility, Expert Systems with Applications 79 (2017) 269–281. doi:10.1016/j.eswa.2017.02.047.

[38] S. Kang, M. Kim, J. Chae, A closed loop based facility layout design using a cuckoo search algorithm, Expert Systems with Applications 93 (2018) 322–335. doi:10.1016/j.eswa.2017.10.038.

[39] A. Klausnitzer, R. Lasch, Optimal facility layout and material handling network design, Computers & Operations Research 103 (2019) 237–251. doi:10.1016/j.cor.2018.11.002.

[40] Y. Xiao, Y. Xie, S. Kulturel-Konak, A. Konak, A problem evolution algorithm with linear programming for the dynamic facility layout problem—a general layout formulation, Computers & Operations Research 88 (2017) 187–207. doi:10.1016/j.cor.2017.06.025.

[41] J. O. Ejeh, S. Liu, L. G. Papageorgiou, Optimal multi-floor process plant layout with production sections, Chemical Engineering Research and Design 137 (2018) 488–501. doi:10.1016/j.cherd.2018.07.018.

[42] A. Ahmadi, M. R. Akbari Jokar, An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems, Applied Mathematical Modelling 40 (9–10) (2016) 5605–5620. doi:10.1016/j.apm.2016.01.014.

[43] H. Navidi, M. Bashiri, M. Messi Bidgoli, A heuristic approach on the facility layout problem based on game theory, International Journal of Production Research 50 (6) (2012) 1512–1527. doi:10.1080/00207543.2010.550638.

[44] B. Javadi, F. Jolai, J. Slomp, M. Rabbani, R. Tavakkoli-Moghaddam, An integrated approach for the cell formation and layout design in cellular manufacturing systems, International Journal of Production Research 51 (20) (2013) 6017–6044. doi:10.1080/00207543.2013.791755.

[45] S. A. Fahmy, Mixed integer linear programming model for integrating cell formation, group layout and group scheduling, in: 2015 IEEE International Conference on Industrial Technology (ICIT), Vol. 38, IEEE, 2015, pp. 2403–2408. doi:10.1109/icit.2015.7125452.

[46] V. R. Ghezavati, A new stochastic mixed integer programming to design integrated cellular manufacturing system: A supply chain framework, International Journal of Industrial Engineering Computations 2 (3) (2011) 563–574. doi:10.5267/j.ijiec.2011.03.003.

[47] S. A. Fahmy, Optimal design and scheduling of cellular manufacturing systems: An experimental study, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Vol. 38, IEEE, 2016, p. 004532–004537. doi:10.1109/smc.2016.7844945.

[48] A. Taghavi, A. Murat, A heuristic procedure for the integrated facility layout design and flow assignment problem, Computers & Industrial Engineering 61 (1) (2011) 55–63. doi:10.1016/j.cie.2011.02.011.

[49] R. Kia, A. Baboli, N. Javadian, R. Tavakkoli-Moghaddam, M. Kazemi, J. Khorrami, Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealing, Computers & Operations Research 39 (11) (2012) 2642–2658. doi:10.1016/j.cor.2012.01.012.

[50] FICO, FICO Xpress Optimization, version 9.2. Accessed: 2024-11-05 (2024).
URL https://www.fico.com/en/products/fico-xpress-optimization

[51] Visual Components Oy, VisualComponents, version 4.9. Accessed: 2024-11-05 (2024).
URL https://www.visualcomponents.com/