Computers & Operations Research 140 (2022) 105674

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Balancing and scheduling assembly lines with human-robot collaboration tasks

Amir Nourmohammadi [a,*], Masood Fathi [a,b], Amos H.C. Ng [a,b]

[a] Division of Intelligent Production Systems, University of Skövde, P.O. Box 408, SE-541 28 Skövde, Sweden
[b] Division of Industrial Engineering and Management, Uppsala University, PO Box 534, Uppsala 75121, Sweden

ARTICLE INFO

Keywords:
Assembly line balancing
Human-robot collaboration
Multiple humans and robots
Joint tasks
Mathematical model
Meta-heuristic

ABSTRACT

In light of the Industry 5.0 trend towards human-centric and resilient industries, human-robot collaboration (HRC) assembly lines can be used to enhance productivity and workers' well-being, provided that the optimal allocation of tasks and available resources can be determined. This study investigates the assembly line balancing problem (ALBP), considering HRC. This problem, abbreviated ALBP-HRC, arises in advanced manufacturing systems, where humans and collaborative robots share the same workplace and can simultaneously perform tasks in parallel or in collaboration. Driven by the need to solve the more complex assembly line-balancing problems found in the automotive industry, this study aims to address the ALBP-HRC with the cycle time and the number of operators (humans and robots) as the primary and secondary objective, respectively. In addition to the traditional ALBP constraints, the human and robot characteristics, in terms of task times, allowing multiple humans and robots at stations, and their joint/collaborative tasks are formulated into a new mixed-integer linear programming (MILP) model. A neighborhood-search simulated annealing (SA) is proposed with customized solution representation and neighborhood search operators designed to fit into the problem characteristics. Furthermore, the proposed SA features an adaptive neighborhood selection mechanism that enables the SA to utilize its exploration history to dynamically choose appropriate neighborhood operators as the search evolves. The proposed MILP and SA are implemented on real cases taken from the automotive industry where stations are designed for HRC. The computational results over different problems show that the adaptive SA produces promising solutions compared to the MILP and other swarm intelligence algorithms, namely genetic algorithm, particle swarm optimization, and artificial bee colony. The comparisons of human/robot versus HRC settings in the case study indicate significant improvement in the productivity of the assembly line when multiple humans and robots with collaborative tasks are permissible at stations.

1. Introduction

Industry 4.0 represents a solid ambition for innovation and further technological development (De Nul et al., 2021) and has received great attention from many industries, such as the automotive and electronics industries. Driven by the growing role of computer-controlled automation technologies that began with Industry 3.0 (such as industrial robots with a high level of precision, repeatability, and strength), manufacturing companies have increased the productivity of their assembly systems by investing in a high degree of automation (Weckenborg et al., 2020). However, many manufacturers cannot fully automate every task in their assembly systems, as the robot technology is often incapable of performing them with the same level of human intelligence and skill. Consequently, they still find the manual tasks performed by human workers as a crucial source of flexibility and adaptability (Weckenborg and Spengler, 2019).

In-line with the Industry 4.0 techno-economic vision, the design of assembly lines in industries has shifted from the traditional configuration, namely either manual or robotic, towards human-robot collaboration (HRC) configurations so as to achieve both higher productivity and flexibility (Dalle Mura and Dini, 2019). In addition, the use of collaborative robots in manufacturing has gained greater attention as one of the key enabling technologies for Industry 4.0. Unlike the previous generations of industrial robots which had to be placed in isolated cells, these robots are designed to work closely next to humans at the same workplace, based on embedded interaction, sensing, and safety

* Corresponding author.
E-mail address: amir.nourmohammadi@his.se (A. Nourmohammadi).

https://doi.org/10.1016/j.cor.2021.105674
Received 7 May 2021; Received in revised form 24 October 2021; Accepted 12 December 2021
Available online 18 December 2021

technologies (Çil et al., 2020). As a result, HRC has recently emerged as a hybrid production environment, aiming to eliminate possible drawbacks of manual or robotic assembly lines while enhancing the productivity, flexibility, and reconfigurability of assembly lines. Fig. 1 shows the 3D layouts of an assembly line with (a) manual, (b) robotic, and (c) HRC environments.

An HRC assembly line can allocate human and robotic resources to execute a production plan sharing the tasks according to the capabilities of the available resources; this is not achievable with monotonically manual or robotic lines (De Nul et al., 2021). Moreover, the close collaborations between humans and robots can enhance not only the productivity of the assembly processes but also the workers' well-being, e.g., by giving them a sense of empowerment and higher work satisfaction (Fletcher et al., 2020). In light of moving towards the human-centric and resilient industries proposed by the emerging Industry 5.0 vision that complements and extends the hallmark features of Industry 4.0, the increasing use of such types of HRC-based collaborative assembly systems is anticipated in the not-so-distant future (De Nul et al., 2021; Demir et al., 2019). Nevertheless, to make all these advantages attainable, efficient line configurations and production plans must be prerequisites so as to make optimal use of the available resources (both humans and robots).

In spite of the well-established literature on the assembly line balancing problem (ALBP), efficient design, configuration, and planning methods for advanced assembly lines with humans and robots considering real-world challenges have yet to be tackled by researchers and industrial practitioners. A few studies have recently dealt with the configuration of HRC assembly lines. For example, (Dalle Mura and Dini, 2019) and (Weckenborg et al., 2020) studied ALBP with HRC where tasks could be performed collaboratively by human and robot. However, the majority of the studies have assumed that only one human and one robot can collaborate at each station, while in real-world settings, multiple humans and robots can share the same workplace to ensure a higher level of flexibility and productivity. As the literature review section shows, there is a lack of research, where multiple humans and robots are permissible at each station to perform the task collaboratively and in parallel. Additionally, the minimization of the number of humans and robots besides the cycle time has rarely been explored.

Originally motivated by the real-world challenges in automotive manufacturing, this study attempts to mitigate the above gaps by dealing with a new ALBP with HRC (abbreviated by ALBP-HRC). Given a known number of stations (*NS*), ALBP-HRC aims to minimize the cycle time (*CT*) and the number of operators (including humans and robots) as the primary and secondary objectives, respectively. Apart from the precedence relationships among tasks, new real-world constraints arising from HRC are considered, namely: (1) the humans' and robots' features while performing tasks, in terms of the capability and/or the task times; (2) allowing multiple humans and robots at each station to perform the assembly tasks, either in parallel or collaboratively, and (3) the collaborative tasks performed jointly by both human and robot at stations, e.g., because these tasks need HRC on two or more shared tasks performed on the same workpiece.

To address the above ALBP-HRC, a new mixed-integer linear programming (MILP) model is proposed, enabling optimal solutions for small-sized problems. The ALBP-HRC, as the extension of the simple ALBP (which is NP-hard), is computationally difficult to optimally solve using exact methods. Therefore, meta-heuristic algorithms seem to be an effective method to solve large-sized instances of ALBP-HRC. To this aim, a novel SA is also developed to address the ALBP-HRC. The proposed SA incorporates new solution representation, feasible solution generation, and neighborhood search mechanisms customized to the characteristics of ALBP-HRC. To improve the efficiency of the SA, an adaptive neighborhood selection mechanism is embedded into the SA (called adaptive SA) that utilizes the history of the algorithm performance to dynamically update the selection probability of the neighborhood searches as the search evolves.

Furthermore, the efficiency of adaptive SA is compared with the MILP and other meta-heuristics, namely genetic algorithm, particle swarm optimization, and artificial bee colony, while addressing real-world cases and generated test problems. Moreover, additional analyses investigate the effect of different scenarios, i.e., only human, only robot, one human, and one robot, and two humans and two robots per station, on the quality of solutions in terms of the considered performance measures.

The main contributions of this study are summarized as follows.

- An ALBP-HRC from a real-world problem is studied.
- A new MILP model is proposed for the problem.
- An adaptive SA algorithm is developed with customized representation, feasible solution generation, and neighborhood searches.
- The performance of adaptive SA is validated by solving real-world cases taken from the automotive industry containing new constraints, namely multiple humans and robots and collaborative tasks.

The remainder of this paper is organized as follows. Section 2 reviews the ALBP-HRC literature. The problem description and a real-world case are explained in Section 3. Section 4 presents the mathematical formulation of the ALBP-HRC. The developed adaptive SA and its elements are described in Section 5. The computational study in Section 6 contains the experimental settings, computational results, sensitivity analyses of results, and managerial insights. Finally, concluding remarks and future research directions are summarized in Section 7.

2. Literature review on assembly line balancing problem with human-robot collaboration (ALBP-HRC)

In this section, the literature most relevant to this study is reviewed. Ding et al., (2014) proposed a procedure/guideline for assigning the tasks between humans and robots in the assembly process of a programmable logic controller module station; they attempted to improve the *CT* as a measure of the production throughput. Michalos et al. (2018)

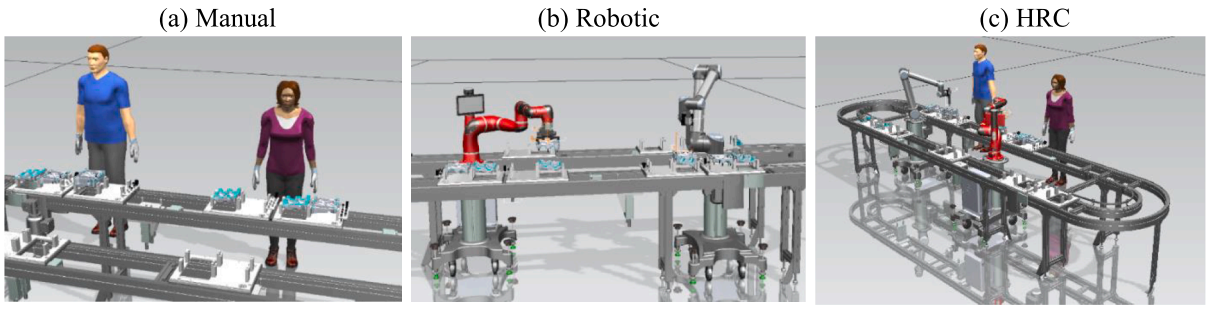


Fig. 1. The 3D layout of an assembly line with (a) manual, (b) robotic, (c) human-robot collaboration (HRC) environment.

proposed a framework based on multi-criteria decision-making for assigning tasks to either a human or a robot at stations given a production assembly sequence. The assignment of tasks to operators was performed for the automotive cell layout of a real case, considering ergonomic and productivity measures. Bruno and Antonelli, (2018) considered task assignment to a human and robot, considering the tasks' features, as well as the skills and capabilities of the human and robot. They proposed a classification approach based on using the existing knowledge regarding a group of classified tasks as the training set and tested their approach on a real industrial case.

Dalle Mura and Dini, (2019) were among the first authors to practically address an ALBP-HRC. The authors developed a genetic algorithm (GA) for optimizing the number of workers, ergonomic loads of humans, and the number of equipment, including cobots. They assumed that one human and one robot could be assigned to each station and tested the performance of the GA on a real case study. Dianatfar et al., (2019) provided a framework for task assignments between human and robot in a diesel engine assembly line while considering collaborations between operators (and their characteristics). In another study, Weckenborg and Spengler, (2019) considered the assembly line costs in terms of the respective numbers of stations, humans, and robots (to be minimized), based on human and robot characteristics, allowing only one human and one robot at stations, and the maximum workloads on human workers. The problem was formulated as a mathematical optimization model, which then was used to solve a small test problem.

Weckenborg et al., (2020) solved an ALBP-HRC in which the *CT* was optimized given an *NS* while considering human and robot characteristics (in terms of task times) and collaborative tasks between a maximum of one human and one robot per station. A mathematical model and a GA-based approach were proposed to address the generated test problems. In another study, Çil et al., (2020) addressed an ALBP-HRC by optimizing *CT* s for different product variants of mixed-models. The authors considered the precedence relationships among tasks and/or models as well as human and robot capabilities (in terms of task times), while assuming that multiple humans and single robots were allowed at stations, but that no collaborative tasks could be performed by a human and robot in parallel. The authors proposed a mathematical model and two meta-heuristics, namely the bee algorithm and artificial bee colony algorithm, to address the generated test problems. Yaphiar and Nugraha (2020) proposed a mathematical model for a mixed-model ALBP-HRC in which, given the *CT* and *NS*, the total cost of an assembly line, in terms of the numbers of humans and robots, was minimized. A mathematical model was proposed in which human and robot capabilities and collaborative tasks were considered, where each station could be equipped with a maximum of one human and one robot. Raatz et al., (2020) addressed task scheduling in an HRC environment, aiming to improve *CT* as the main measure. They proposed a framework based on a GA that considered human and robot features and other factors (including human ergonomic and safety factors) to obtain optimal schedules for an industrial case concerning a gearbox assembly line.

Recently, Li et al., (2021b) considered a cost-oriented ALBP-HRC in which the *CT* and total cost of operators including humans and robots were optimized. The authors assumed that different types of robots were available, and that only one robot was allowed at each station. They also hypothesized that humans and robots could not work in parallel at stations, owing to safety aspects. The problem was formulated as a mathematical model, and a meta-heuristic known as "migration bird optimization" was proposed for solving the generated test problems. Gualtieri et al., (2021) proposed a systematic method to switch from a manual to a collaborative assembly station where only one human and one robot were allowed to perform the assembly tasks of a cash register machine either separately, in parallel, or collaboratively. Koltai et al., (2021) dealt with HRC by assuming that a maximum of one human and one robot could be assigned to specific stations with shared workplaces where only sequential and parallel (not collaborative) tasks were allowed to avoid interference among operators. The author proposed a mathematical formulation of the problem and optimized a power inverter assembly line.

A summary of the literature review on ALBP-HRC is given in Table 1 based on categorizing the studies from different perspectives, including the type of problem, objective, constraints, and solution approach. The last row shows the contributions of the present study to the ALBP-HRC literature. This study is among the first attempts to address the ALBP-HRC using a real case study where, aside from the *CT*, the respective numbers of humans and robots are considered for optimization. Furthermore, in addition to the ALBP constraints, human and robot characteristics in terms of task times, multiple humans and robots are allowed at stations. Moreover, joint tasks collaboratively performed by a human and robot are considered. A new MILP model and an adaptive version of SA with customized mechanisms are developed to address this problem.

3. Problem description

This study is originated from a real-world ALBP from the automotive industry, in which decision-makers (DM) have attempted to enhance the flexibility and agility of their assembly lines by adopting HRC technology in the era of Industry 4.0. This can be achieved by taking advantage of both human and robot strengths while (re)designing the advanced assembly lines where humans and robots work in parallel or collaboratively. The assembly lines are equipped with stop/control checks, from which information between robots and humans is shared continuously on a real-time basis. Moreover, the robots are equipped with detection sensors to enable a safe HRC environment.

Fig. 2 shows a snapshot of the assembly line from a case study with one human and one robot per station. There are 28 tasks performed in two stations to assemble a mass balancing system (MBS), as shown in Fig. 3. The task times for the human and robot are shown on the top left and top right of each task node/rectangles, respectively. The four highlighted tasks (i.e., 5 and 6 in green and 7 and 8 in brown) show the joint tasks that need to be performed by both humans and robots collaboratively. In this case, the robot supports the human by performing heavy/repetitive lifting of the upper part of the MBS. In contrast, the human simultaneously guides the upper part of the MBS held by the robot arm toward the proper assembly area on the lower part of the MBS. These tasks require the collaboration of the human and robot on the shared workpiece and enhance the human's working conditions.

Aside from the traditional limitations of the ALBP (e.g., the precedence relationships among tasks), the new real-world challenges raised by HRC must be considered. The DM aims to determine an optimized level of human-automation collaborations, i.e., through the efficient utilization of humans and robots. Overall, three decision problems must be jointly dealt with to address the considered ALBP-HRC: (1) assigning tasks to stations, (2) assigning resources (i.e., humans and robots) to stations, and (3) scheduling tasks at stations and between resources. The aim is to minimize the *CT* and numbers of humans and robots as primary and secondary objectives, respectively. The problem assumptions are as follows:

- A single model of a product is assembled in a straight line.
- Humans and robots have different skills and capabilities, as indicated through different task times known in advance.
- Each station can be equipped with a maximum number of humans and a maximum number of robots, also known in advance.
- Humans and robots can perform tasks separately, in parallel, or collaboratively. In the latter type, the tasks are indicated as joint tasks (e.g., the tasks highlighted in Fig. 3), where both humans and robots perform them in collaboration.
- The robots are always available, and their maintenance/breakdown operations are not considered.

Based on the MILP model presented in the next section, a feasible

Table 1
Summary of the literature review on assembly line balancing problem with human-robot collaboration (ALBP-HRC).

Study	Problem				Objective			Constraint				Approach		
	Task to station assignment	Task to operator assignment	Task scheduling	Real case	Cycle time	Number of stations	Number of humans	Human ergonomic load	Number of robots	Precedence relationships	Human & robot capability	Multiple humans and robots	Collaborative/ joint tasks	Mathematical model	Meta-heuristic algorithm	Framework/ guideline
(Ding et al., 2014)		X		X	X						X					X
(Michalos et al., 2018)		X		X	X			X			X					X
(Bruno and Antonelli, 2018)		X		X						X	X		X			X
(Dalle Mura and Dini, 2019)	X	X	X	X			X	X	X	X	X		X		X	
(Dianatfar et al., 2019)		X		X	X						X		X			X
(Weckenborg and Spengler, 2019)	X	X	X			X	X		X	X	X			X		
(Weckenborg et al., 2020)	X	X	X		X					X	X		X	X	X	
(Çil et al., 2020)	X	X	X		X					X	X			X	X	
(Yaphiar et al., 2020)	X	X	X				X		X	X	X		X	X		
(Raatz et al., 2020)	X	X	X	X	X			X		X	X				X	X
(Li et al., 2021b)	X	X	X		X		X		X	X	X			X	X	
(Gualtieri et al., 2021)		X	X	X	X					X	X		X			X
(Koltai et al., 2021)	X	X	X	X	X					X	X			X		
This study	X	X	X	X	X		X		X	X	X	X	X	X	X	

Fig. 2. Assembly line of a case study with two stations and one human and one robot per station.

solution for the above real case is obtained and is illustrated in Fig. 4. The horizontal rectangles show the tasks and their assignments among the two stations with one human and one robot per station and the task schedules. The results show that $ct = 58.2$ with the total of two humans and two robots. Notably, in this solution, tasks 5 and 6, as well as 7 and 8, as joint tasks are scheduled to be performed collaboratively by humans and robots, as shown by the dashed rectangles. Moreover, there is an unavoidable idle time for the robot at station 1; this is caused by the precedence relationships among the tasks.

4. Mathematical model of ALBP-HRC

In this section, the mathematical formulation of the ALBP-HRC is presented. A straight assembly line is considered, where a set of tasks (assembly operations) $i = 1, \cdots, NT$ have to be assigned to an ordered number of stations, i.e., $j = 1, \cdots, NS$, and can be performed by a specified number of humans ($l = 1, \cdots, NH$) and robots ($r = 1, \cdots, NR$), in parallel/jointly. The processing time of task i when performed by either the human or robot is determined by considering their respective skills and capabilities. The considered ALBP-HRC attempts to minimize the *CT* as the primary objective and the respective total numbers of humans and robots as the two secondary objectives, given the *NS*. Moreover, aside from the precedence relationships among tasks, the real-world situations arising from the automotive case study are considered, i.e., multiple humans and robots per station, and joint tasks collaboratively performed by humans and robots. The notations used in the problem formulation

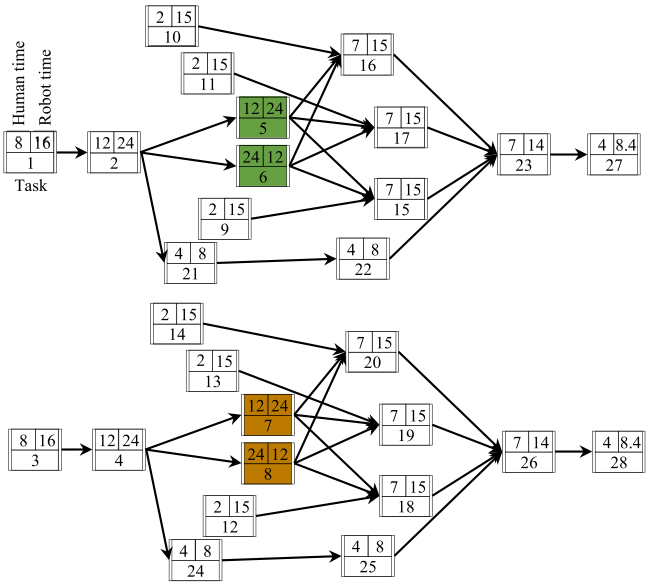


Fig. 3. Precedence relationships among tasks in the case study.

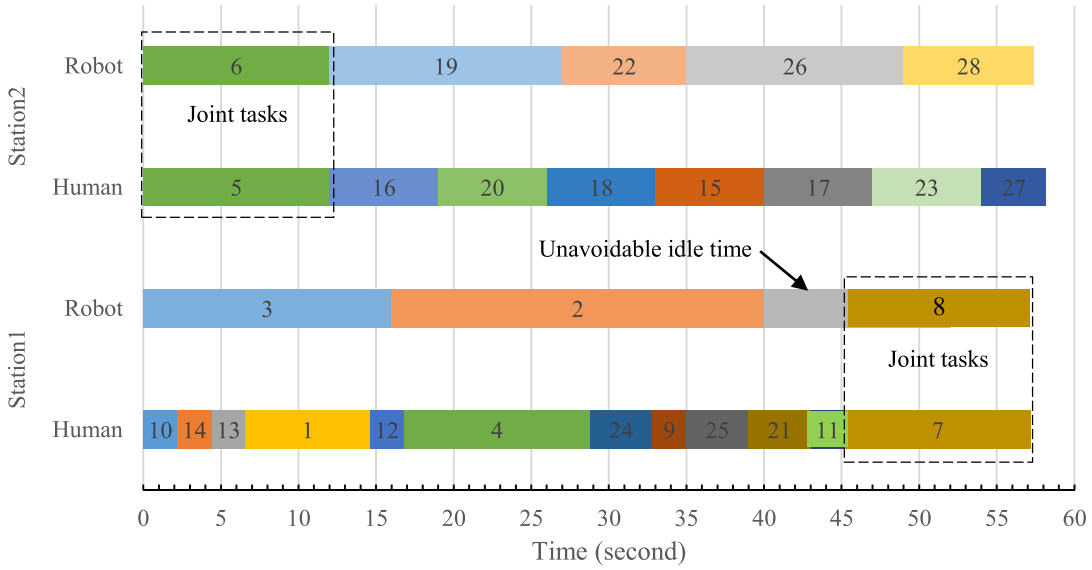


Fig. 4. A feasible solution for assembly line balancing problem with HRC (ALBP-HRC) in the case study.

are given in Table 2.

The proposed MILP for the ALBP-HRC is given as follows:

$$Minimize\ of\ v = ct + (\sum_{j=1}^{NS}\sum_{k=1}^{NH} hu_{jk} + \sum_{j=1}^{NS}\sum_{r=1}^{NR} ro_{jr})/((NH+NR)\times NS+1) \quad (1)$$

$$\sum_{j=1}^{NS}\sum_{k=1}^{NH} x_{ijk} + \sum_{j=1}^{NS}\sum_{r=1}^{NR} y_{ijr} = 1; \forall i \quad (2)$$

$$\sum_{j=1}^{NS}\sum_{k=1}^{NH} x_{hjk}\times j + \sum_{j=1}^{NS}\sum_{r=1}^{NR} y_{hjr}\times j \le \sum_{j=1}^{NS}\sum_{k=1}^{NH} x_{ijk}\times j + \sum_{j=1}^{NS}\sum_{r=1}^{NR} y_{ijr}\times j;\ \forall i, \forall h \in \{p_i\} \quad (3)$$

$$t_i = \sum_{j=1}^{NS}\sum_{k=1}^{NH} th_i \times x_{ijk} + \sum_{j=1}^{NS}\sum_{r=1}^{NR} tr_i \times y_{ijr}; \forall i \quad (4)$$

$$st_i - st_h + M\times\left(1-\sum_{k=1}^{NH} x_{hjk} - \sum_{r=1}^{NR} y_{hjr}\right) + M\times\left(1-\sum_{k=1}^{NH} x_{ijk} - \sum_{r=1}^{NR} y_{ijr}\right) \ge t_h; \forall i, \forall j, \forall h \in \{p_i\} \quad (5)$$

$$st_f - st_i + M\times(1-x_{fjk}) + M\times(1-x_{ijk}) + M\times(1-u_{if}) \ge t_i; \forall i, \forall j, \forall k, \forall f \notin \{psall_i\} \quad (6)$$

$$st_i - st_f + M\times(1-x_{fjk}) + M\times(1-x_{ijk}) + M\times u_{if} \ge t_f; \forall i, \forall j, \forall k, \forall f \notin \{psall_i\} \quad (7)$$

$$st_f - st_i + M\times(1-y_{fjr}) + M\times(1-y_{ijr}) + M\times(1-v_{if}) \ge t_i; \forall i, \forall j, \forall r, \forall f \notin \{psall_i\} \quad (8)$$

$$st_i - st_f + M\times(1-y_{fjr}) + M\times(1-y_{ijr}) + M\times v_{if} \ge t_f; \forall i, \forall j, \forall r, \forall f \notin \{psall_i\} \quad (9)$$

$$\sum_{i=1}^{NT} x_{ijk} \le NT \times hu_{jk}; \forall j, \forall k \quad (10)$$

$$\sum_{i=1}^{NT} y_{ijr} \le NT \times ro_{jr}; \forall j, \forall r \quad (11)$$

$$\sum_{k=1}^{NH} hu_{jk} \le NH; \forall j \quad (12)$$

$$\sum_{r=1}^{NR} ro_{jr} \le NR; \forall j \quad (13)$$

$$hu_{jk+1} \le hu_{jk}; \forall j, \forall k \le NH \quad (14)$$

$$ro_{jr+1} \le ro_{jr}; \forall j, \forall r \le NR \quad (15)$$

$$st_i + t_i \le co_i; \forall i \quad (16)$$

$$co_i \le cos_j + M\times(1-\sum_{k=1}^{NH} x_{ijk} - \sum_{r=1}^{NR} y_{ijr}); \forall i, \forall j \quad (17)$$

$$cow_j \le ct; \forall j \quad (18)$$

$$st_i = st_h; \forall i, h \in \{jt_i\} \quad (19)$$

$$\sum_{k=1}^{NH} x_{ijk} + \sum_{k=1}^{NH} x_{hjk} \le 1; \forall i, \forall j, h \in \{jt_i\} \quad (20)$$

$$\sum_{r=1}^{NR} y_{ijr} + \sum_{r=1}^{NR} y_{hjr} \le 1; \forall i, \forall j, h \in \{jt_i\} \quad (21)$$

Table 2
Notations used in ALBP-HRC formulation.

Notation	Definition
Indices:	
i, h, f, d	set of tasks ($i = 1, \cdots, NT$)
j, g	set of stations ($j = 1, \cdots, NS$)
l, k	set of humans ($l = 1, \cdots, NH$)
r	set of robots ($r = 1, \cdots, NR$)
Parameters:	
NT	number of tasks
NS	given number of stations
NH	maximum number of humans that can be assigned to each station
NR	maximum number of robots that can be assigned to each station
th_i	time of task i if performed by human
tr_i	time of task i if performed by robot
p_i	set of immediate predecessors of task i
$pall_i$	set of all predecessors of task i
s_i	set of immediate successors of task i
$sall_i$	set of all successors of task i
$psall_i$	set of all predecessors and successors of task i
jt_i	set of joint tasks for task i
rpw_i	ranked positional weight of task i; $rpw_i = \sum_{h \in pall_i} \min(th_h, tr_h)$
M	a large number
Decision variables:	
Binary:	
x_{ijk}	$\begin{cases} 1; \text{if task } i \text{ is assigned to human } k \text{ in station } j \\ 0; \text{otherwise} \end{cases}$
y_{ijr}	$\begin{cases} 1; \text{ if task } i \text{ is assigned to robot } r \text{ in station } j \\ 0; \text{otherwise} \end{cases}$
u_{ih}	$\begin{cases} 1; \text{if task } i \text{ is executed before task h in the sequence of tasks assigned to the same human} \\ 0; \text{otherwise} \end{cases}$
v_{ih}	$\begin{cases} 1; \text{if task } i \text{ is executed before task h in the sequence of tasks assigned to the same robot} \\ 0; \text{otherwise} \end{cases}$
hu_{jk}	$\begin{cases} 1; \text{if human } k \text{ is usedin station} j \\ 0; \text{otherwise} \end{cases}$
ro_{jr}	$\begin{cases} 1; \text{if robot } r \text{ is usedin station} j \\ 0; \text{otherwise} \end{cases}$
Real positive:	
ct	cycle time
t_i	processing time of task i after being assigned to either human or robot
st_i	start time of task i
co_i	completion time of task i
cos_j	completion time of all tasks assigned to station j

$$\sum_{k=1}^{NH} x_{ijk} + \sum_{r=1}^{NR} y_{ijr} = \sum_{k=1}^{NH} x_{hjk} + \sum_{r=1}^{NR} y_{hjr}; \forall i, \forall j, h \in \{jt_i\} \tag{22}$$

$$x_{ijk}, y_{ijr}, u_{ih}, v_{ih}, hu_{jk}, ro_{jr} \in \{0, 1\}; \forall i, \forall h, \forall j, \forall k, \forall r \tag{23}$$

$$ct, t_i, st_i, co_i, cow_j \in \mathbb{R}^+; \forall i, \forall h, \forall j, \forall k, \forall r \tag{24}$$

In objective Function (1), the first term corresponds to the cycle time (ct), whereas the second term represents the total number of operators, i.e., the humans and robots, as weighted by a coefficient $((NH + NR) \times NS + 1)^{-1}$ which guarantees that the second term is always less than 1. For this reason, Function (1) minimizes the ct as the primary objective, and the numbers of humans and the number of robots as the secondary objective. Constraint (2) guarantees that each task is assigned to either one human or one robot. Constraint (3) ensures that all the precedence relations among tasks are satisfied. Constraint (4) determines the time of each task after it has been assigned to either a human or a robot. Constraints (5) to (9) schedule tasks at each station and among humans and robots, depending on their precedence relationships; i.e., Constraint (5) represents if a task is the immediate predecessor of another task; Constraint (6) represents if, e.g., tasks i and f are not directly/indirectly connected, considering all precedence relationships, but i is performed before f by the same human or vice versa; Constraint (7) considers tasks not directly/indirectly connected, but where task f is performed before i by the same human; whereas Constraints (8) and (9) similarly schedule tasks likewise, if performed by the same robot. Constraint (10) states that if a task is assigned to human k in station j, then wk_{jk} must be equal to 1, as well (the worker is utilized at that station). Constraint (11) resembles Constraint (10), but from a robot perspective. Constraints (12) and (13) guarantee that the numbers of workers and robots assigned to each station do not exceed the maximum possible numbers of humans and robots at each station, respectively. Constraints (14) and (15) ensure that the humans' and robots' assignments to each station are performed in increasing order of their indexes. Constraint (16) determines the completion time of each task. Constraint (17) specifies the completion time of the entire set of tasks assigned to a specific station. Constraint (18) sets the ct as the maximum of the completion times of all stations. Constraints (19)–(22) address joint tasks by ensuring that they are started simultaneously while being performed by a human and a robot at the same station. Finally, Constraints (23) and (24) define the domain of the decision variables.

5. Proposed simulated annealing (SA) algorithms for ALBP-HRC

Inspired by the annealing process in metals, the SA algorithm is a local search-based heuristic algorithm developed by Kirkpatrick et al., (1983). Since then, the SA algorithm has shown an excellent capability to comprehensively explore the search spaces of optimization problems

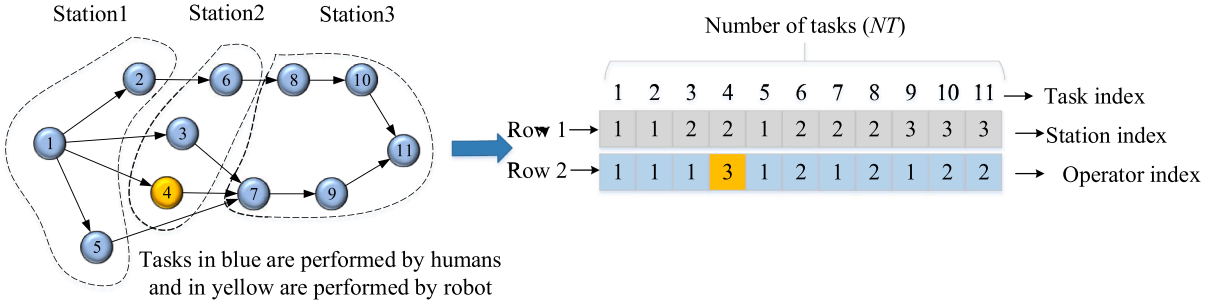


Fig. 5. Solution representation of simulated annealing (SA).

through stochastic searching and to escape from local optima by accepting worse solutions. The SA algorithm starts with an initial solution (so_0) at the initial temperature (T_0). Then, at each temperature (T), a neighbor solution (*nso*) based on current solution (*cso*) is generated using a neighborhood search. If the *nso* has a lower/equal objective value (in minimization objective) than the *cso* (i.e., $f(nso) \leq f(cso)$), then the *nso* is selected to replace the *cso*. Otherwise, the *nso* is accepted with a probability of $exp^{-(f(nso)-f(cso))/T}$ (known as the Boltzmann function). The neighborhood search is repeated iteratively for a maximum number of iterations (*maxIT*), after which T is updated according to a cooling rate ($cr; cr \in (0,1)$) using a cooling scheme (i.e., $T \times cr$). Notably, while T decreases, the probability of accepting worse solutions is reduced. The above procedure is repeated until the SA algorithm stops, i.e., when T reaches the final temperature (T_f).

The proposed SA is developed based on the SA suggested by Yu et al., (2021) with customized features adapted to address the ALBP-HRC. The SA incorporates an adaptive selection mechanism that dynamically tracks the history of the neighborhood searches to guide the algorithm through selecting appropriate neighborhoods during the search progress. The following subsections present the detailed descriptions of the elements of the adaptive SA, namely: (1) solution representation, (2) feasible solution generation, (3) initial solution generation, (4) neighborhood search, and (5) neighborhood selection. The same notations introduced in Table 2 are used in the following sections.

5.1. Solution representation

The solution representation is composed of two rows. The first row shows the station index of each task, which ranges between (1,…,*NS*). The second row shows the operator index for performing each task, which can range between (1,…,*NH*) if performed by a human or between (*NH* + 1,…,*NH* + *NR*) if performed by a robot. The length of this representation (the number of columns) equals the number of tasks (*NT*). In Fig. 5, an 11-task problem with the shown precedence graph and specific assignments of tasks to stations and operators is illustrated. The corresponding representation of the specific solution to this problem is also shown. In this example, station 1 has only one human, station 2 includes one human and one robot, and station 3 has two human operators to perform the assigned tasks.

5.2. Feasible solution generation

Considering the three decision problems mentioned above while addressing the ALBP-HRC, the first two decisions are defined, namely, the assignment of tasks to the stations and the assignment of tasks to operators, i.e., both humans and robots are defined through the first and the second rows in the solution representation as shown in Fig. 5. However, the third decision, namely, the scheduling of tasks within stations and among assigned operators in terms of the start and completion times of tasks considering the time variations between humans and robots, remains to be determined. Thus, to decode the solution representation to a feasible solution for the ALBP-HRC, a feasible solution generation mechanism is proposed in Algorithm 1.

The algorithm starts with a first station, in which a set of tasks and operators are assigned to it given a solution (*so*). Next, a candid list of tasks (with either assigned predecessors or without any predecessors) is created, and their start and completion times are determined considering the allocated operators. Then, a task is chosen using a roulette wheel selection based on their ranked positional weights (rpw_i). Next, the start time and the completion time for the selected task (and its joint task, if any) are set. Accordingly, the operators' completion times and candid list of tasks are updated. The above mechanism is repeated until the candid list of tasks at the current station is empty. Then a new station is opened, and the above mechanism is repeated. Finally, the algorithm stops when all stations are considered; at this point, a feasible solution is generated.

Algorithm1: The feasible solution generation

1. **Input:** $so, NT, NS, NH, NR, th_i, tr_i, p_i, rpw_i, jt_i$;
2. $stc = 0$; initialize the station counter;
3. **While** $stc \leq NS$
4. $stc = stc + 1$; open a new station
5. $stop$ = set of operators assigned to the current station;
6. $opct = 0$; for $stop$ initialize the operators' completion times
7. clt = set of the candid list of tasks in the current station with assigned predecessors or without any predecessors;
8. **While** $clt \neq \emptyset$
9. Calculate the start times of clt for $stop$ considering $opct$;
10. Calculate the completion times of clt for $stop$ considering lines 10;
11. slt = a task chosen from clt using the roulette wheel selection based on rpw_i; tasks with higher rpw_i have a higher likelihood to be selected;
12. **If** $jt_{slt} = \emptyset$; slt does not have a joint task
13. Set the start time of slt considering line 9;
14. Set the completion time of slt considering line 10;
15. **Else**; slt has a joint task $slt2$ ($jt_{slt} = slt2$)
16. Set the start time of slt and $slt2$ considering the maximum of their start times in line 9;
17. Set the completion time of slt and $slt2$ considering line 16;
18. **End**
19. Update clt, $opct$ and feasible solution;
20. **End**
21. **Output:** the feasible solution

5.3. Initial solution generation

To generate an initial solution, first, a lower bound on ct (ct_{low}) is calculated. This lower bound is theoretically calculated by assuming that *NH*+*NR* operators are available at each station and that the processing times of all tasks can be considered as equal to the minimum task times either by a human or a robot ($t_i = \min(th_i, tr_i)$). Then, a theoretical ct_{low} can be calculated as follows:

$$ct_{low} = \max\left\{\left\lceil \frac{\sum_{\forall i} t_i}{((NH+NR)\times NS)} \right\rceil, max_{\forall i}(t_i)\right\} \quad (25)$$

The procedure in Algorithm 2 is used to generate an initial solution for the ALBP-HRC. In this procedure, the coefficient τ is a controlling

parameter for determining the maximum workload in terms of the task times at the first $(NS-1)$ *th* stations. To find better initial solutions, different values of τ are tested, and the best solution found is chosen as the initial solution.

Algorithm2: The initial solution generation

1. **Input:** $ct_{low}, NT, NS, NH, NR, th_i, tr_i, p_i, rpw_i, jt_i$;
2. $stc = 0$; initialize the station counter
3. **While** $stc \leq NS$
4. $stc = stc + 1$; open a new station
5. $stop$ = set of all operators assigned to the current station; $\|stop\| = NH + NR$ ($\|\|$ is the size of set)
6. $opct = 0$; for $stop$ initialize the operators' completion times
7. clt = set of the candid list of tasks with assigned predecessors or without any predecessors;
8. **While** $clt \neq \emptyset$
9. Calculate the start times of clt for $stop$ considering $opct$;
10. Calculate the completion times of clt for $stop$ considering lines 9;
11. **If** $stc \leq NS$-1
12. Exclude tasks in clt with all $stop$'s completion times $> \tau \times ct_{low}$; τ is a coefficient ($\tau \geq 1$)
13. **If** clt=∅
14. Go to line 4;
15. **end**
16. **End**
17. slt = a task chosen from clt using the roulette wheel selection based on rpw_i; tasks with higher rpw_i have a higher likelihood to be selected;
18. **If** $jt_{slt} = \emptyset$; slt does not have a joint task
19. Assign slt to $stop$ with lower completion time;
20. Set the start time of slt considering line 9;
21. Set the completion time of slt considering line 10;
22. **Else**; slt has a joint task $slt2$ ($jt_{slt} = slt2$)
23. Assign slt and $slt2$ to $stop$ with lower completion times and to be performed jointly by a human and a robot or vice versa
24. Set the start time of slt and $slt2$ considering the maximum of their start times in line 9;
25. Set the completion time of slt and $slt2$ considering line 24;
26. **End**
27. Update clt, $opct$ and initial solution;
28. **End**
29. **Output:** the initial solution

5.4. Neighborhood search

The neighborhood searches are responsible for the algorithm's progression, based on generating neighbor solutions from the current solution. Considering the proposed solution representation for the ALBP-HRC, four types of neighborhood searches are introduced to create neighborhood solutions while preventing the generation of unfeasible solutions, as follows:

- ***Station-swap***: This neighborhood search (briefly named *st-sw*) applies to a station row and is based on swapping the stations of two randomly selected tasks, as described below:
  Select two tasks i and h randomly at different stations with no precedence relationships as j_i and j_h.
  Determine the first and the last stations to which the selected tasks can be moved, namely Fj_i and Lj_i for task i, and Fj_h and Lj_h for task h, by considering the stations of their immediate predecessors and successors.
  If none of the tasks i and j have no joint tasks, then:
    If $j_i < j_h$ and $j_i \geq Fj_h$ and $j_h \leq Lj_i$, then swap the stations, i.e., j_i and j_h.
    If $j_i > j_h$ and $j_i \leq Lj_h$ and $j_h \geq Fj_i$, then swap the stations, i.e., j_i and j_h.
  Otherwise, if tasks i and j have a joint task such as d, then:
    If $j_i < j_h$ and $j_i \geq Fj_h$ and $j_h \leq Lj_i$, then swap the stations, i.e., j_i, j_h and j_d.
    If $j_i > j_h$ and $j_i \leq Lj_h$ and $j_h \geq Fj_i$, then swap the stations, i.e., j_i, j_h and j_d.

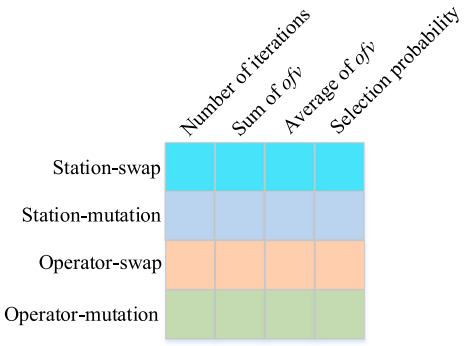


Fig. 6. The adaptive selection probability matrix.

- ***Station-mutation:*** This neighborhood search (*st-mu*) applies to station row, and is based on switching the station of a randomly selected task, as described below:
  Randomly select a task i, where its current station j_i can be changed between a first station Fj_i and the last station Lj_i to which it can be moved.
  If task i has no joint task, then:
    If $j_i = Fj_i$ and $j_i < Lj_i$, select a new random station for i where $j_i \in rand(Fj_i, Lj_i]$; "(" means not inclusive and "[" means inclusive of values.
    If $j_i > Fj_i$ and $j_i = Lj_i$, select a new random station for i where $j_i \in rand[Fj_i, Lj_i)$.
    If $j_i > Fj_i$ and $j_i < Lj_i$, select a new random station for i where $j_i \in \{rand[Fj_i, Lj_i] - j_i\}$.
  Otherwise, if task i has a joint task such as d, then:
    If $j_i = Fj_i$ and $j_i < Lj_i$, select a new random station for both i and d where $j_i, j_d \in rand(Fj_i, Lj_i]$.
    If $j_i > Fj_i$ and $j_i = Lj_i$, select a new random station for both i and d where $j_i, j_d \in rand[Fj_i, Lj_i)$.
    If $j_i > Fj_i$ and $j_i < Lj_i$, select a new random station for both i and d where $j_i, j_d \in \{rand[Fj_i, Lj_i] - j_i\}$.
- ***Operator-swap:*** This neighborhood search (*op-sw*) applies to the operator row, and is based on swapping the operators for two randomly selected tasks, as described below:
  Select two tasks i and h randomly with different operators as o_i and o_h.
  If none of tasks i and h have joint tasks, then swap their operators, i.e., o_i and o_h.
  Otherwise, if both task i and h are joint tasks ($i \in jt_h$), then swap their operators, i.e., o_i and o_h.
  If one of tasks i or j has different joint task such as d, then:
    If $\{(o_i \leq NH)$ and $(o_h > NH)\}$ or $\{(o_i > NH)$ and $(o_h \leq NH)\}$, then go to step 1.
    Otherwise, swap the operators of tasks i and h, i.e., o_i and o_h.
- ***Operator-mutation:*** This neighborhood search (*op-mu*) applies to the operator row, and is based on switching the operator of a randomly selected task, as described below:
  Randomly select a task i with the operator o_i.
  Select a new operator o_i' for task i randomly from $o_i' \in \{[1, NH + NR] - o_i\}$.
  If task i has no joint task, then switch its operator as $o_i = o_i'$.
  Otherwise, if task i has a joint task such as d with the operator o_d, then:
    If $\{(o_i' \leq NH)$ and $(o_d \leq NH)\}$ or $\{(o_i' > NH)$ and $(o_d > NH)\}$, then go to step 1.
    Otherwise, switch the operator of tasks i to the new operator, i.e., $o_i = o_i'$.

5.5. Neighborhood selection

In addition to the neighborhood search mechanisms, the choices of neighborhood searches can significantly influence the performance of the SA algorithm while addressing the ALBP-HRC. The conventional selection strategy would be to select one/more neighborhood searches randomly and then evaluate the efficiency of new solutions. However, since the performance of SA is highly dependent on the choice of neighborhood, a dynamic neighborhood selection can enhance the efficiency of SA while avoiding pure stochastic selections. Thus, a new scheme is introduced, named adaptive neighborhood selection, that utilizes the exploration history of the SA algorithm to choose appropriate neighborhood searches for forthcoming local searches. Initially, at T_0, the selection probability of the neighborhood searches is equal. As the SA algorithm proceeds, the performance of the neighborhood searches in terms of the objective function values (*ofv*) of the obtained solutions are stored and exploited to update the selection probability of the neighborhood searches. To preserve the information of the neighborhood searches, a matrix structure is defined and updated, as shown in Fig. 6. The rows of this matrix show different neighborhood searches, whereas the columns indicate different measures (described as follows). The first column shows the total number of iterations where each neighborhood is applied. The second column represents the cumulative sum of *ofv* as obtained by each neighborhood search. If any neighborhood has not yet been applied, the number of iterations is set to 1, and the sum of *ofv* is set to the worst *ofv* found so far. The third column calculates the average values of *ofv* obtained by each neighborhood search. Finally, the fourth column calculates the selection probability for each neighborhood search by dividing its average of *ofv* by the total sum of column three i.e., the sum of all average values of *ofv*.

Considering the previous mechanisms in Sections 5.1–5.5, a novel SA algorithm is proposed for ALBP-HRC as presented in Algorithm 3.

Algorithm3: The proposed SA for ALBP-HRC

1. **Input:** Parameters of problem (ct_{low},NT,NS,NH, NR, th_i, tr_i p_i, rpw_i, jt_i) and SA (T_0,T_f, cr, $maxIT$)
2. Generate an initial solution (so_0) using section 5.3;
3. Set current solution cso=so_0, the best solution bso=so_0, and the worst solution wso=so_0;
4. **While** $T > T_f$
5. Initialize the neighborhood selection matrix in Fig. 6 and neighborhoods' selection probabilities to ¼;
6. **For** $iter = 1, \cdots, maxIT$
7. Considering the current neighborhoods' selection probabilities, select a random neighborhood search;
8. Considering the selected neighborhood search in line 7, generate a new neighborhood solution (nso) using section 5.4;
9. Generate a feasible solution for nso using section 5.2;
10. Update the neighborhood selection matrix; i.e., columns 1 and 2 based on adaptive selection mechanism;
11. $\Delta = f(nso) - f(cso)$;
12. **If** $\Delta \leq 0$
13. Update current solution $cso = nso$;
14. **Else if** $rand < \exp(-\Delta/T)$
15. Update current solution $cso = nso$;
16. **End**
17. **If** $f(nso) < f(bso)$
18. Update $bso = nso$;
19. **Else if** $f(nso) > f(wso)$
20. Update $wso = nso$;
21. **End**
22. **End**
23. Update the neighborhood selection matrix; i.e., columns 3 and 4 based on adaptive selection mechanism;
24. Update temperature $T_{next} = T_{current} \times cr$;
25. **End**
26. **Output:** Report the best solution (bso) found

6. Computational study

In this section, the performances of the proposed MILP model and adaptive SA in addressing different problem sizes and instances as well as real-world case studies, are evaluated. The computational study is structured to discuss experimental settings in Section 6.1, followed by parameter tuning in Section 6.2. In Section 6.3, the computational results are presented and discussed. Sections 6.4 and 6.5 provide sensitivity analysis on neighborhood search and selection, respectively, while Section 6.6 compares human/robot versus HRC settings. Finally, Section 6.7 outlines the managerial insight.

6.1. Experimental setting

The experimental study presented in this section was raised from a real-world automotive manufacturing problem. The real problem consisted of different case studies, one of which was already presented in Section 3. The rest of the case studies are not described, owing to space limitations. The specific data related to all four case studies, including the precedence relationships, human and robot characteristics (in terms of task times), joint tasks, and maximum numbers of humans and robots at stations, were provided by the company as the input data, and are available in the Supplementary material.

To further compare the performances of the proposed approaches on different ALBP-HRC instances, some test problems were generated. To do so, benchmark problems existing in the ALBP literature (available at https://assembly-line-balancing.de/) were considered, and the missing data were generated and added to them. The generated test problems were categorized into three problem sizes considering the number of tasks (NT), i.e., $NT \leq 11$ as small, $11 < NT \leq 30$ as medium, and $NT \geq 70$ as large. While generating the test problem data, humans' task times were assumed to be equal to the original task times, whereas the task times for robots were assumed to be twice the human times. Furthermore, the tasks that were allowed to be performed in parallel according to the given precedence relationships were considered as joint tasks. Moreover, for the sets of joint tasks, humans' and robots' task times were assumed to be the same so that their start and completion times could be scheduled in parallel. The generated test problems are available in the Supplementary material.

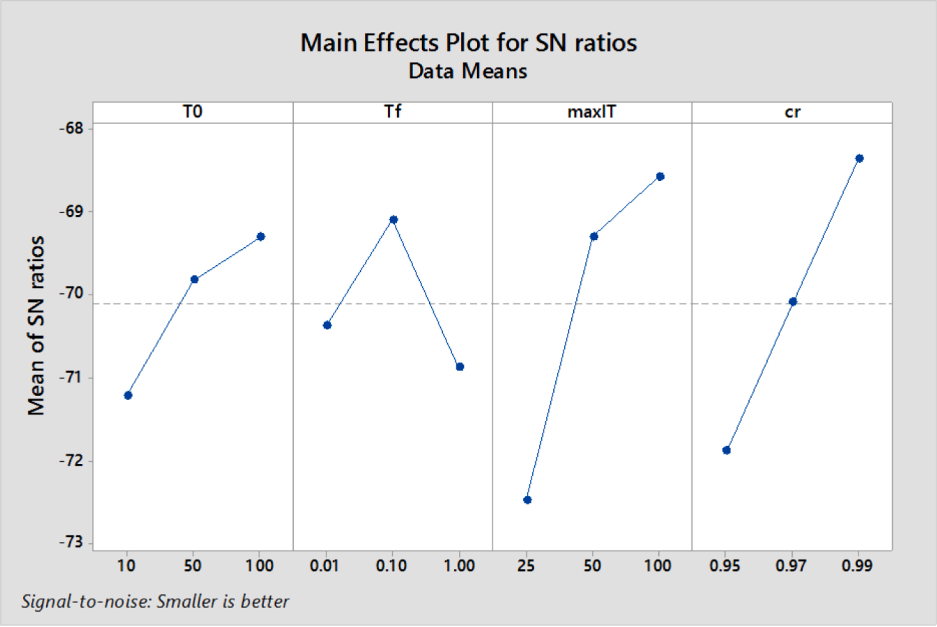
To solve this case study and the generated test problems using the newly formulated MILP, the CPLEX solver of the "General Algebraic Modeling System" software (v34.02) was used on a Core i9 PC with a 3.10 GHz processor and 64 GB of RAM. The solver was terminated either when the optimal solution was found or when the CPU time reached 1800 s. The numbers of variables, constraints, and nonzero coefficients of the CPLEX model for all problems are reported in the Supplementary material. It can be observed that as the dimensionality of the problem increases in terms of NT, NS, NH, NR, the respective number of variables, constraints, and nonzero coefficients of the CPLEX model significantly increase.

To measure the efficiency of the proposed SA algorithm, the results are compared with MILP as well as three swarm intelligence algorithms, namely genetic algorithm (GA), particle swarm optimization (PSO), and artificial bee colony (ABC). It is worth mentioning that these algorithms are extended based on the GA proposed by Fathi et al., (2019), PSO developed by Petropoulos and Nearchou (2011), and ABC proposed by Li et al., (2021a). All algorithms are coded in MATLAB R2020b and run on the same PC. The stopping conditions of SA was set to when the final temperature (T_f) was reached, while GA, PSO and ABC were stopped when a maximum number of generations was attained. Due to the stochastic nature of the meta-heuristic algorithms, the best solutions found after five runs of each algorithm are reported.

6.2. Parameter tuning

To ensure the high performance of the proposed SA algorithm, finding the best value of the algorithm's parameters is essential. In this study, a well-known design of experiments method named Taguchi design (Taguchi et al., 2005) is used for parameter setting. After

(a)



(b)

Fig. 7. SA parameter tuning by Taguchi: (a) main effect plots for means, (b) main effect plots for S/N ratios.

considering several values for each parameter, this method sets a reasonable combination of parameter values to select the best parameter levels so that a certain measure known as signal to noise (S/N) ratios is maximized. The SA parameters and their levels considered in this study are; (1) initial temperature (T_0 = 10,50,100), (2) final temperature (T_f = 0.01,0.1,1.0), (3) maximum iteration ($maxIT$ = 25,50,100) and (4) cooling rate (cr = 0.95,0.97,0.99). The $L_9(3^4)$ design of experiments given by the Taguchi method was used for SA parameter tuning where the average of five runs for solving 25% of the problems was considered. The main effects' diagrams for Means and S/N ratios are plotted in Fig. 7 (a) and 7(b), respectively. According to these figures, the higher gradients of the Means and S/N ratios diagrams for $maxIT$ and ct suggest that they have a more significant effect on SA's performance compared to T_0 and T_f.

Similarly, the GA, PSO, and ABC parameters are tuned using the Taguchi method, and the results of parameter selections for all of these algorithms are shown in Table 3.

6.3. Computational results

In this section, the computational results of the MILP model SA, GA, PSO, and ABC are compared by addressing the different problems. Tables 4–7 show the best results obtained by the different approaches for the case studies and the different problem sizes. In these tables, the first six columns show the problem characteristics, including the instance number, name, number of tasks (NT), NS, the maximum number of humans (NH), and maximum number of robots (NR) per station. The best results from the different approaches are shown in the following columns. The computational results are given in terms of the (near) optimal OFV, CPU time (CPU), and relative gap (Gap). The gaps for the

Table 3
Parameter settings for GA, PSO, and ABC algorithms.

	SA	GA	PSO	ABC
Parameter setting	$T0 = 100$	$popsize = 100$	$popsize = 100$	$popsize_{employed} = 100$
	$Tf = 0.1$	$maxG = 1000$	$maxG = 1000$	$popsize_{onlookers} = 100$
	$maxIT = 100$	$pr_{crossover} = 0.7$	$c1 = 2.0$	$maxG = 500$
	$cr = 0.99$	$pr_{mutation} = 0.01$	$c2 = 2.0$	$limit = 20$
		$pr_{elitism} = 0.29$	$Inertia_0 = 0.9$	
			$Inertia_{rate} = 0.95$	

MILP are reported from the CPLEX solver results, whereas the related gaps for different algorithms are calculated using Equation (26). Thus, a zero-gap shows that the MILP results are reached, a positive gap shows worse results than the MILP, and a negative gap shows better results than the MILP.

$$Gap_{Alg} = \frac{(OFV_{Alg} - OFV_{MIP})}{OFV_{MIP}} \times 100; Alg = SA, GA, PSO, ABC \quad (26)$$

From Table 4, it can be observed that the MILP results for Case 2 are optimal, whereas for Cases 1 and 3, the solutions have positive gaps. However, for Case 4, which is a large-size problem, no solution can be found within the 1800 s. Among the algorithms, SA shows the best performance, as the average gap from the MILP solution equals 0.35%, relative to 18.32%, 29.52% and 47.19% for GA, PSO and ABC, respectively. Notably, SA could obtain the same solutions, or even in two instances (i.e., no. 1 and 5), better solutions than the MILP.

From Table 5, it is noticeable that the MILP results for all small-sized instances are optimal. In addition, SA could obtain the same solution as the MILP in all 17 instances. However, GA and PSO have the same gap of 9.67% from optimal solutions, followed by ABC with an average gap of 9.73%. Therefore, the performance of SA in addressing small-sized problems is optimal.

In Table 6, it can be observed that the MILP results for 8 out of 18 instances are optimal, whereas, for the rest of the instances, the gaps are positive, as the MILP could not reach optimal solutions within the 1800 s. SA shows the best performance (12 out of 18 equal to MILP) with an average gap of 0.05%, followed by GA, PSO, and ABC, with average gaps of 10.90%, 14.25%, and 19.99%, respectively.

From Table 7, it can be observed that the MILP can find no feasible solutions for the large-sized instances within the 1800 s. Among the tested algorithms, SA shows the best performance, as the average *OFV* for all instances (i.e., 4429.8) is smaller than the respective values from GA, PSO, and ABC (i.e., 5209.1, 6097.4 and 7600.9, respectively).

Considering the average computational time over all problems solved, MILP has the highest CPU time, followed by SA, GA, PSO, and ABC.

6.4. Sensitivity analysis of neighborhood selection

To explore how adaptive selection can contribute to the quality of solutions, in this section, an analysis of random and variable neighborhood selection strategies compared to the adaptive SA (ASA) proposed in Section 5.5 is performed. These selection mechanisms are described below.

- ***Random***: This mechanism uses a random number generated from one to four to select which type of neighborhood searches in Section 5.4 will be used to generate a new solution at each iteration of SA.

Table 4
Case studies best experimental results.

No	Problem	NT	NS	NH	NR	MILP			SA			GA			PSO			ABC		
						OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)
1	Case 1	28	2	1	1	59.0	1800.0	14.32	58.8	196.9	−0.34	61.2	30.4	3.73	62.0	24.7	5.08	66.8	24.5	13.22
2				2	2	35.1	1800.0	7.22	35.1	154.6	0.00	38.1	36.3	8.55	40.2	30.5	14.50	46.3	29.4	31.92
3	Case 2	19	3	1	1	56.6	108.2	0.00	56.6	77.0	0.00	56.7	20.3	0.25	56.7	17.0	0.25	56.7	18.6	0.25
4				2	2	56.3	292.2	0.00	56.3	115.1	0.00	61.4	25.3	9.02	62.5	20.2	10.93	72.4	21.2	28.55
5	Case 3	41	4	1	1	53.5	1800.0	24.74	52.5	400.8	−1.87	55.0	43.0	2.80	58.2	39.6	8.79	60.9	40.6	13.83
6				2	2	29.2	1800.0	4.81	30.5	473.8	4.30	54.3	79.3	85.60	69.5	44.6	137.58	86.4	45.5	195.37
7	Case 4	88	9	1	1	-*	1800.0	–	63.3	574.9	–	57.9	116.7	–	68.5	117.3		90.1	113.6	
8				2	2	–	1800.0	–	56.8	478.5	–	50.5	148.8	–	59.2	143.0		74.0	157.7	
Avg						48.3	1400.1	8.51	51.2	309.0	0.35	54.4	62.5	18.32	59.6	54.6	29.52	69.2	56.4	47.19

*No solution found during 1800 s.

Table 5
Small-sized problems best experimental results.

No	Problem	NT	NS	NH	NR	MILP			SA			GA			PSO			ABC		
						OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)
9	Mertens	7	2	1	1	11.6	0.6	0.00	11.6	0.8	0.00	11.8	13.7	1.72	11.8	7.1	1.72	11.8	8.9	1.72
10				2	2	10.4	0.9	0.00	10.4	0.1	0.00	10.4	13.8	0.00	10.4	9.7	0.00	10.4	11.4	0.00
11		7	3	1	1	9.6	0.5	0.00	9.6	0.4	0.00	9.6	29.9	0.00	9.6	26.9	0.00	9.6	20.2	0.00
12				2	2	6.5	0.3	0.00	6.5	1.4	0.00	6.5	19.7	0.00	6.5	11.2	0.00	6.5	23.0	0.00
13		7	4	1	1	6.7	0.5	0.00	6.7	48.3	0.00	6.7	29.0	0.00	6.7	25.6	0.00	6.7	28.7	0.00
14	Jaeschke	9	2	1	1	19.6	0.2	0.00	19.6	0.4	0.00	19.8	11.9	1.02	19.8	9.6	1.02	19.8	8.3	1.02
15				2	2	14.6	0.6	0.00	14.6	3.0	0.00	19.6	14.8	34.35	19.6	9.3	34.35	19.6	11.6	34.35
16		9	3	1	1	13.6	0.2	0.00	13.6	0.9	0.00	19.6	12.6	44.21	19.6	9.1	44.21	19.6	9.1	44.21
17				2	2	10.5	0.5	0.00	10.5	3.1	0.00	11.5	18.1	9.56	11.5	12.3	9.56	11.5	11.9	9.56
18		9	4	1	1	10.6	0.9	0.00	10.6	0.7	0.00	10.6	19.0	0.00	10.6	10.1	0.00	10.6	12.6	0.00
19				2	2	9.4	0.9	0.00	9.4	2.9	0.00	9.4	40.2	0.00	9.4	43.3	0.00	9.4	41.1	0.00
20	Jackson	11	3	1	1	13.6	0.7	0.00	13.6	7.8	0.00	14.6	17.1	7.37	14.6	12.8	7.37	14.7	12.7	8.42
21				2	2	9.5	0.8	0.00	9.5	23.8	0.00	9.5	19.1	0.00	9.5	12.7	0.00	9.5	15.4	0.00
22		11	4	1	1	10.7	1.2	0.00	10.7	1.2	0.00	14.6	19.8	36.46	14.6	11.8	36.46	14.6	13.6	36.46
23				2	2	8.4	2.4	0.00	8.4	0.6	0.00	8.4	42.2	0.00	8.4	35.1	0.00	8.4	23.1	0.00
24		11	5	1	1	9.5	4.7	0.00	9.5	3.4	0.00	9.5	20.2	0.00	9.5	12.2	0.00	9.5	16.1	0.00
25				2	2	6.4	3.3	0.00	6.4	49.8	0.00	8.3	45.5	29.63	8.3	42.1	29.63	8.3	42.6	29.63
Avg.						10.6	1.1	0.00	10.6	8.7	0.00	11.8	22.7	9.67	11.8	17.7	9.67	11.8	18.2	9.73

Table 6
Medium-sized problems best experimental results.

No	Problem	NT	NS	NH	NR	MILP			SA			GA			PSO			ABC		
						OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)
26	Mitchell	21	5	1	1	18.6	9.3	0.00	18.6	81.3	0.00	18.7	25.1	0.49	18.7	23.1	0.49	18.7	23.1	0.49
27				2	2	16.4	14.8	0.00	16.4	154.4	0.00	18.4	34.2	12.21	18.4	39.3	12.21	18.4	46.0	12.21
28			6	1	1	16.6	6.2	0.00	16.6	72.6	0.00	16.6	30.4	0.00	16.7	20.7	0.46	17.7	32.0	6.48
29				2	2	15.4	19.8	0.00	15.4	218.7	0.26	18.3	79.4	19.27	18.3	88.6	19.27	18.3	86.0	19.27
30			7	1	1	14.6	6.7	0.00	14.6	84.5	0.00	16.5	34.6	13.24	16.5	73.0	13.24	16.6	67.4	13.70
31				2	2	14.3	74.8	0.00	14.3	150.7	0.00	18.3	79.9	27.71	18.3	92.1	27.71	18.3	86.9	27.71
32	Heskia	28	3	1	1	240.9	620.4	0.00	240.9	181.8	0.00	242.9	28.0	0.83	258.9	28.4	7.47	284.9	27.5	18.27
33				2	2	180.5	455.2	0.00	180.6	118.5	0.04	232.5	35.8	28.80	232.6	32.3	28.85	232.6	29.7	28.85
34			4	1	1	181.9	1800.0	1.16	181.9	156.2	0.00	196.9	30.8	8.25	231.9	25.4	27.49	232.9	26.2	28.04
35				2	2	132.6	1800.0	0.09	132.6	256.3	0.04	144.5	40.1	9.01	156.6	89.6	18.10	160.5	35.6	21.07
36			5	1	1	148.9	1800.0	2.69	148.9	204.0	0.00	156.9	35.3	5.37	164.9	30.4	10.74	205.9	30.6	38.28
37				2	2	124.5	1800.0	0.04	124.5	231.2	0.00	124.5	42.4	0.04	132.5	35.6	6.47	146.5	71.9	17.71
38	Sawyer	30	5	1	1	46.9	1800.0	11.05	46.9	309.0	0.00	48.8	39.0	4.07	49.9	28.2	6.40	50.9	34.2	8.53
39				2	2	38.5	1800.0	0.37	38.6	506.5	0.25	41.5	42.1	7.80	41.5	32.5	7.92	41.5	38.9	7.92
40			7	1	1	34.9	1800.0	0.38	34.9	452.5	0.00	41.8	40.7	19.89	41.9	34.8	20.08	48.8	35.9	39.96
41				2	2	28.4	1800.0	0.24	28.5	583.0	0.12	30.5	42.9	7.15	32.4	43.1	14.06	33.4	46.0	17.58
42			8	1	1	30.9	1800.0	0.38	30.9	338.2	0.00	34.8	42.3	12.57	35.7	30.3	15.62	37.8	38.0	22.29
43				2	2	25.4	1800.0	0.12	25.5	535.5	0.24	30.4	112.3	19.55	30.5	65.9	19.90	33.4	79.8	31.47
Avg.						72.8	1067.1	0.92	72.8	257.5	0.05	79.6	45.3	10.90	84.2	45.2	14.25	89.8	46.4	19.99

Table 7
Large-sized problems best experimental results.

No	Problem	NT	NS	NH	NR	MILP			SA			GA			PSO			ABC		
						OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)	OFV	CPU	Gap (%)
44	Tonge	70	8	1	1	-*	1800.0	–	324.9	546.0	–	324.9	82.8	–	376.9	86.8	–	555.9	102.2	–
45				2	2	–	1800.0	–	215.8	839.7	–	242.5	99.2	–	369.5	94.1	–	430.5	126.6	–
46			9	1	1	–	1800.0	–	286.9	503.9	–	290.9	89.4	–	418.9	79.1	–	569.9	92.1	–
47				2	2	–	1800.0	–	215.8	436.3	–	226.5	99.0	–	245.5	112.6	–	288.5	141.2	–
48			10	1	1	–	1800.0	–	279.0	489.3	–	417.9	82.8	–	516.9	80.0	–	670.9	110.9	–
49				2	2	–	1800.0	–	208.7	463.8	–	215.5	102.7	–	223.5	113.5	–	242.5	113.4	–
50	Arcus1	83	9	1	1	–	1800.0	–	7091.9	972.4	–	7091.9	122.1	–	7377.9	100.7	–	11679.9	109.6	–
51				2	2	–	1800.0	–	5264.8	491.5	–	5430.5	132.0	–	7091.5	128.7	–	8541.5	142.0	–
52			10	1	1	–	1800.0	–	7091.9	321.6	–	7378.0	118.3	–	11679.8	115.9	–	11679.9	112.5	–
53				2	2	–	1800.0	–	4737.8	491.2	–	5296.5	342.0	–	5287.5	295.7	–	5513.5	156.9	–
54			11	1	1	–	1800.0	–	7091.8	512.8	–	7091.8	105.6	–	7091.9	102.1	–	9905.9	111.3	–
55				2	2	–	1800.0	–	4494.7	338.0	–	5157.5	348.5	–	5334.4	364.9	–	5443.5	235.0	–
56	Arcus2	111	14	1	1	–	1800.0	–	8692.9	346.3	–	12257.9	195.4	–	18165.9	162.8	–	25518.9	165.2	–
57				2	2	–	1800.0	–	6370.8	613.9	–	7530.5	205.5	–	8342.5	201.2	–	8613.5	233.1	–
58			15	1	1	–	1800.0	–	8587.9	461.9	–	11918.8	165.7	–	13205.9	183.4	–	18255.9	180.4	–
59				2	2	–	1800.0	–	5749.8	1490.4	–	7700.5	216.8	–	7781.5	278.6	–	8373.5	262.6	–
60			16	1	1	–	1800.0	–	7341.9	459.5	–	7840.9	199.1	–	8288.8	164.5	–	12267.9	165.5	–
61				2	2	–	1800.0	–	5689.7	1159.6	–	7350.4	393.4	–	7954.5	230.1	–	8264.5	386.8	–
Avg.						–	1800.0	–	4429.8	607.7	–	5209.1	172.2	–	6097.4	160.8	–	7600.9	163.7	–

*No solution found during 1800 s.

This procedure results in a variant of SA with random neighborhood selection abbreviated by RSA.

- ***Variable:*** In this type of selection, two consecutive neighborhood searches are chosen randomly to perform a nested neighborhood search on the current solution. As this selection first chooses a random neighborhood search to perform another random search consecutively, it is called variable neighborhood selection. This type of selection forms a new variant of the SA with variable neighborhood selection abbreviated by VSA.

The gaps of the best solutions found by all variants of SA (namely, ASA, RSA, and VSA) from the (near) optimal solutions found by all approaches over different sizes and related problems are shown in Fig. 8. According to this figure, ASA finds the (near) optimal solutions for most of the problems.

Moreover, Table 8 shows a pairwise comparison of the number of instances in which worse, equal, and better solutions are obtained by the ASA versus the MILP, RSA, and VSA over the case studies and test problems with different sizes. For instance, in the case study with a total of eight problems, the MILP was superior only in one instance. In contrast, the ASA finds equal or even better solutions in three and four instances, respectively. The following three columns show that for zero, one, and eight instances, worse, equal, or better solutions, respectively, are found by ASA versus RSA. The last three columns indicate that ASA always finds better solutions than VSA. The rest of the information in the table follows the same description. Overall, considering all 61 instances solved, except for seven and one instances where the MILP and VSA dominate ASA, respectively, ASA finds equal and even better solutions than the MILP, RSA, and VSA. This implies that adaptive neighborhood selection has significantly improved the efficiency of SA for solving the ALBP-HRC.

6.5. Sensitivity analysis of neighborhood searches

To further analyze the effects of different neighborhood searches (i.e., *st-sw*, *st-sw*, *op-sw*, and *op-mu*) presented in Section 5.4 on the quality of the solutions, a sensitivity analysis on the neighborhood moves is performed. Table 9 presents the four scenarios considered to investigate how variations on probabilities of selecting the neighborhood searches can contribute to the quality of solutions compared to the adaptive SA. For this analysis, at each scenario, one of the four neighborhood searches is weighted by a higher selection probability of 1/2, while the sum of all probabilities equals one, as shown in Table 9.

To better measure the effect of different neighborhood searches, the four scenarios are performed by RSA over the large-sized problems (instances 44 to 61), and the gaps for the best solution obtained by each scenario from the best solution found by the adaptive SA are illustrated in Fig. 9. It can be observed that in scenario 2, in which the *st-mu* operator has a weighted selection probability of 1/2, a lower gap with an average of 23.54% is achieved. Scenario 1 (with *st-sw*) shows the next lower gap with an average of 48.52%, followed by Scenarios 3 and 4 with the average gaps of 57.29% and 60.11%, respectively. These results imply that the neighborhood searches *st-mu*, and *st-sw*, performed on station level, have a higher effect on the quality of solutions than the neighborhood searches performed on the operator level.

6.6. Human/robot versus HRC analyses

The use of collaborative robots in the manufacturing industry to perform a wide range of assembly tasks either in parallel or with cooperation with humans has paved the way for manufacturers to enhance their flexibility, productivity, and reconfigurability. To further illustrate the benefits of HRC compared to the only human and only robot production environment, the following scenarios are considered:

- ***Human only (H):*** only one human exists at each station.

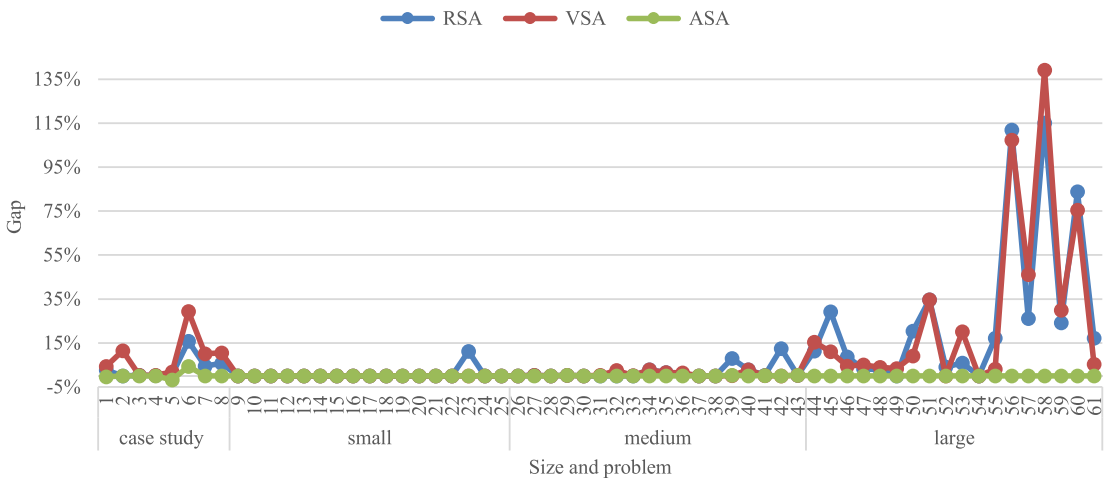


Fig. 8. Best solution gaps of SA variants from (near) optimal solutions over different problems.

Table 8
Comparison of ASA with MILP, RSA, and VSA over different problems.

size	number of instances	ASA vs MILP			ASA vs RSA			ASA vs VSA		
		worse	equal	better	worse	equal	better	worse	equal	better
Case study	8	1	3	4	0	1	8	0	0	8
Small	17	0	17	0	0	17	0	0	17	0
Medium	18	6	12	0	0	9	9	1	7	9
Large	18	0	0	18	0	0	18	0	0	18
Total	61	7	32	22	0	27	35	1	24	35

Table 9
Scenarios for analysis of the neighborhood searches.

Scenario	Neighborhood search			
	st-sw	*st-mu*	*op-sw*	*op-mu*
1	1/2	1/6	1/6	1/6
2	1/6	1/2	1/6	1/6
3	1/6	1/6	1/2	1/6
4	1/6	1/6	1/6	1/2

- ***Robot only (R):*** only one robot exists at each station.
- ***One human and one robot (H-R):*** one human and one robot exist at each station.
- ***Two humans and two robots (2H-2R):*** two humans and two robots (multi-operators) exist at each station.

Fig. 10 (a)-(d) show the solutions of the adaptive SA for real case 1 regarding each scenario. Each diagram represents the assignment of tasks among the stations, the assignment of operators (human, robot, or both) to the stations, and the schedules of tasks for each operator. The results in Fig. 10 (a) and (b) show that the manual (with only one human) and automated (with only one robot) production environment result in *ct* of 102.8 s and 205 s, respectively. For the latter scenario, this is caused by the additional times robots need to analyze the workpiece and process the tasks using the sensing technology compared to the human flexibility in handling tasks. It is remarkable to note that since only one operator (either a human or a robot) exists at each station for H

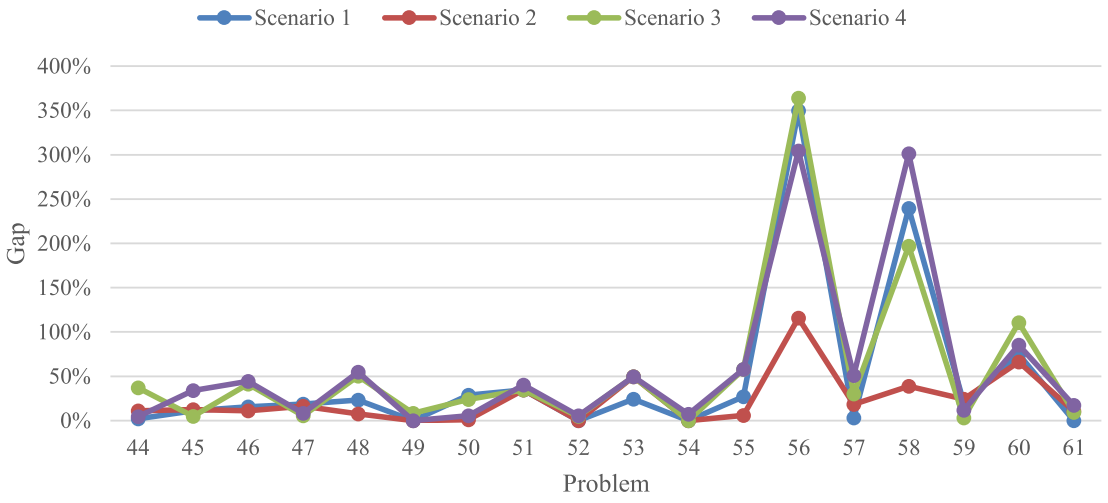
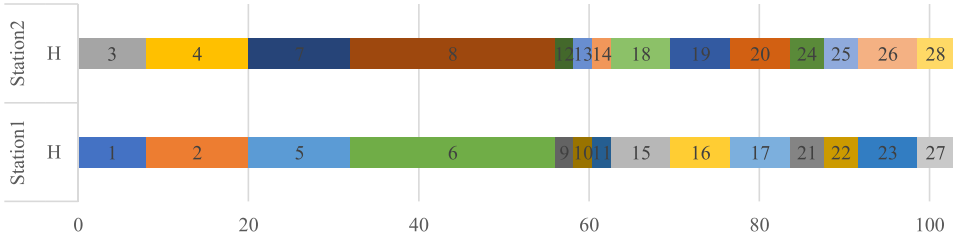
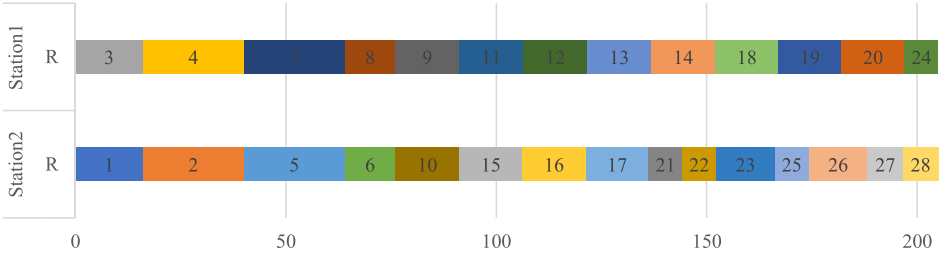


Fig. 9. Comparison of the gaps of four scenarios from adaptive SA.

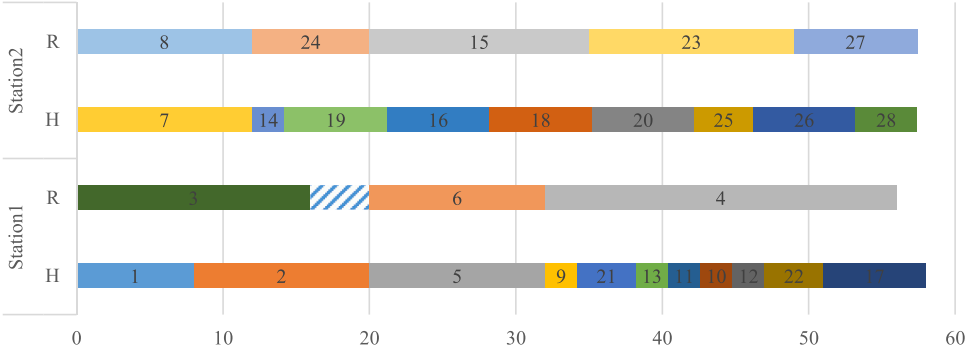
(a) Scenario H: only one human per station with CT=102.8s.



(b) Scenario R: only one robot per station with CT=205s.



(c) Scenario H-R: one human and one robot per station with CT=58s.



(d) Scenario 2H-2R: two humans and two robots per station with CT=34.2s.

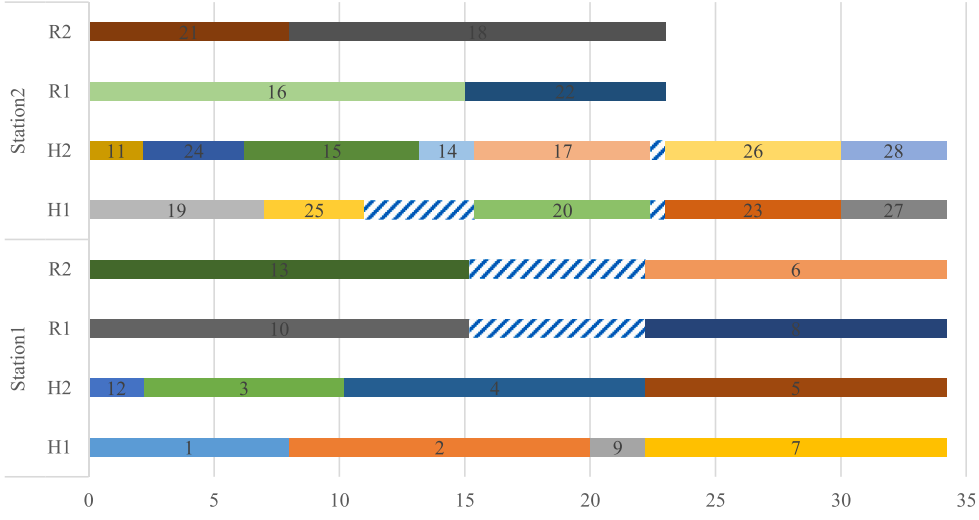


Fig. 10. Comparison of solutions by adaptive SA for case 1 with different scenarios: (a) H, (b) R, (c) H-R, and (d) 2H-2R.

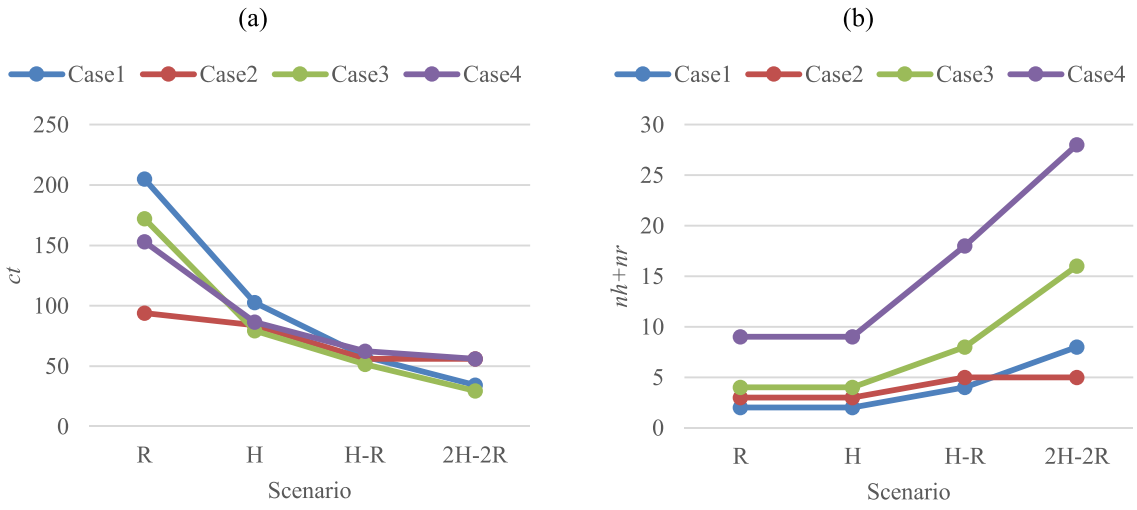


Fig. 11. Comparison of four scenarios for each case in terms of (a) *ct* and (b)$nh + nr$

and R scenarios, no joint tasks are allowed.

Regarding HRC, Fig. 10 (c) illustrates the H-R scenario where the parallel working of one human and one robot and their cooperation in the joint tasks results in a smaller *ct* of 58 s. The robot in the first station has an unavoidable idle time of 4 s until its human counterpart has finished the predecessor task (i.e., task 2). Furthermore, the joint tasks (5, 6) and (7, 8) are performed collaboratively in stations one and two, respectively.

Moreover, Fig. 10 (d) shows the 2 h-2R scenario with multi-humans and multi-robots at each station empowered to perform joint tasks collaboratively, which results in a lower *ct* of 34.2 s. Both robots at the first station, have inevitable idle times of 7 s until the predecessors of tasks 6 and 8, i.e., tasks 2 and 4 are finished, respectively. Similarly, the joint tasks (5,6) and (7,8) are performed through human and robot collaboration in the first station.

Furthermore, Fig. 11 (a) and (b) compare the solutions obtained by the adaptive SA for different scenarios of each real case in terms of *ct* and $nh + nr$, respectively. The diagram in Fig. 11 (a) shows a trend that while moving towards multiple humans and robots at each station working in parallel and collaboratively, the resulting *ct* has decreased for most of the cases. In Fig. 11 (b), the anlaysis of different scenarios regarding the total number of humans and robots (i.e., $nh + nr$) intuitively show an increase in the number of required operators to obtain the associated *ct*.

To further demonstrate the effect of different scenarios on the quality of solutions obtained by the adaptive SA for different problems, Table 10 compares H, R, H-R, and 2H-2R scenarios in terms of *ct*, *nh*, and *nr*. In this table, the first two columns show the characteristics of the problems, namely the name and number of stations (*NS*). The following columns show the best objective functions found for different scenarios. Using this table, the DM can compare how adding operators (human or robot or both) can affect the considered performance measures. For example, in Case 1 scenario H, has the best solution of $ct = 102.8$, nh=2 and$nr = 0$, while in scenario R the best solution has $ct = 205$, nh=0 and$nr = 2$, and so on. One can observe from this table that for scenario R, the *ct* increases given the higher processing times of robots compared to the humans. Another observation is that for the majority of problems, by moving from scenario H and R towards H-R and 2H-2R scenarios, the *ct* decreases while the *nh* and *nr* increase.

Moreover, based on the adaptive SA results, the *ct* gaps associated with the comparison of H-R versus H (only human) and 2H-2R versus H are shown in Fig. 12. In this figure, for case 1 with two stations (shown by Case 1–2), the comparison of the H-R scenario with H shows a gap (improvement) of 43% in cycle time, while the 2H-2R scenario implicates 66% improvement in *ct* compared to H scenario. Considering all case studies, H-R results in 27% to 43% improvements in *ct* and 35% on average. Additionally, 2H-2R improves *ct* between 33% and 67% and 49% on average. For small sizes problems, *ct* improvement by H-R ranges from 9% to 25% and 15% on average, while 2H-2R includes 25% to 40% improvement and 34% on average. In the medium size, H-R has improved the *ct* ranging from 11% to 33% with an average of 24%, while 2H-2R improves *ct* from 12% to 51% and 35% on average. Finally, in the large size problems, H-R improved the *ct* ranging from 1% to 27% with an average of 18%, while 2H-2R could improve the *ct* ranging from 37% to 51% and 42% on average.

6.7. Managerial insight

From the above results, the following managerial implications can be derived.

- The HRC, compared to only human or only robot design, can improve the *ct* depending on the assembly line characteristics in terms of precedence relationships, collaborative tasks, as well as available resources (e.g., number of humans and robots, task times). The more the possibility of HRC at stations with high levels of safety and interactions, the better the managers can make the most out of this emerging hybrid production settings in terms of productivity, flexibility, and (re)configurability.
- Using the proposed SA, managers in the industry can solve the considered ALBP-HRC so that an efficient design, in terms of *ct* and number of operators ($nh + nr$), could be obtained while satisfying the real-world challenges, namely multiple humans and robots and collaborative tasks. Moreover, the analysis of HRC versus human or robot supports the managers in deciding which setting suits their intended design. The H-R scenario (i.e., one human and one robot per station) has been successfully implemented in the first case study from the automotive industry, as shown in Fig. 2. Using the proposed SA, the balancing lead time of the assembly planner could be improved significantly compared to the current planning process, which can be devoted to other tasks such as operator training and development activities.
- Based on the human operators' feedback in the case study, the tasks jointly performed by HRC could improve their working environment by assigning high-risk tasks (e.g., heavy lift, awkward postures) to collaborative robots. Thus, through HRC and work-sharing with

Table 10
Comparison of Adaptive SA results for different problems and scenarios.

Problem	*NS*	*ct*				*nh*				*nr*			
		H	R	H-R	2H-2R	H	R	H-R	2H-2R	H	R	H-R	2H-2R
Case 1	2	102.8	205	58	34.2	2	0	2	4	0	2	2	4
Case 2	3	84	94	56	56	3	0	3	3	0	3	1	1
Case 3	4	79.3	172.2	51.6	29.6	4	0	4	8	0	4	4	8
Case 4	9	86.5	153.2	62.4	56	9	0	9	15	0	9	9	13
Mertens	2	14	26	11	10	2	0	2	3	0	2	1	1
	3	10	16	9	6	3	0	3	5	0	3	1	1
	4	8	16	6	-*	4	0	4	–	0	4	2	–
Jaeschke	2	21	38	19	14	2	0	2	4	0	2	1	1
	3	15	25	13	10	3	0	3	5	0	3	1	1
	4	12	20	10	9	4	0	4	5	0	4	1	1
Jackson	3	15	26	13	9	3	0	3	5	0	3	1	1
	4	12	21	10	8	4	0	4	6	0	4	2	1
	5	10	18	9	6	5	0	5	7	0	5	1	2
Mitchell	5	21	42	18	16	5	0	5	7	0	5	2	1
	6	18	36	16	15	6	0	6	8	0	6	2	2
	7	16	32	14	14	7	0	7	8	0	7	2	1
Heskia	3	356	678	240	180	3	0	3	6	0	3	3	2
	4	267	508	181	132	4	0	4	8	0	4	4	3
	5	214	408	148	124	5	0	5	9	0	5	5	1
Sawyer	5	65	126	46	38	5	0	5	9	0	5	5	3
	7	48	90	34	28	7	0	7	12	0	7	6	2
	8	41	78	30	25	8	0	8	13	0	8	7	3
Tonge	8	441	876	324	215	8	0	8	16	0	8	8	12
	9	394	778	286	215	9	0	9	18	0	9	9	11
	10	355	700	278	208	10	0	10	20	0	10	10	7
Arcus1	9	8859	16,828	7091	5264	9	0	9	17	0	9	9	12
	10	7844	15,172	7091	4737	10	0	10	19	0	10	8	15
	11	7167	14,182	7091	4494	11	0	11	22	0	11	7	11
Arcus2	14	10,855	21,778	8692	6370	14	0	14	24	0	14	13	19
	15	10,159	20,318	8587	5749	15	0	15	29	0	15	14	18
	16	9726	19,452	7341	5689	16	0	16	28	0	16	15	17
Avg.		1848.9	3642.3	1543.1	1125.4	6.5	0.0	6.5	11.4	0.0	6.5	5.0	5.8

*No solution.

collaborative robots, particularly when humans are subjected to tasks with high musculoskeletal disorder risks, the DM can improve the job satisfaction of the operators, which can further enhance the well-being and social aspects of such human-centered production environment. Additionally, integrating other technologies in HRC, such as augmented reality or collision detection, can further support the operators' interactions with collaborative robots in real-time while contributing to Industry 4.0.

- The proposed SA can provide the DM in a manufacturing company with efficient solutions for ALBP-HRC in terms of cycle time and numbers of humans and robots. However, additional cost and benefit analyses of the cycle time improvement versus increasing the number of humans and robots should be performed to find the desired line configuration. Moreover, in this study, the numbers of humans and robots are equally weighted in the secondary objective. However, the difference between humans' and robots' fixed and operational costs during assembly line (re)configuration for HRC is not considered. Thus, a cost-oriented approach towards optimizing ALBP-HRC considering the marginal expenditures of humans and robots should be further studied. Additionally, the inclusion of workers' postural risks while solving ALBP-HRC should be considered to promote humans' well-being further.

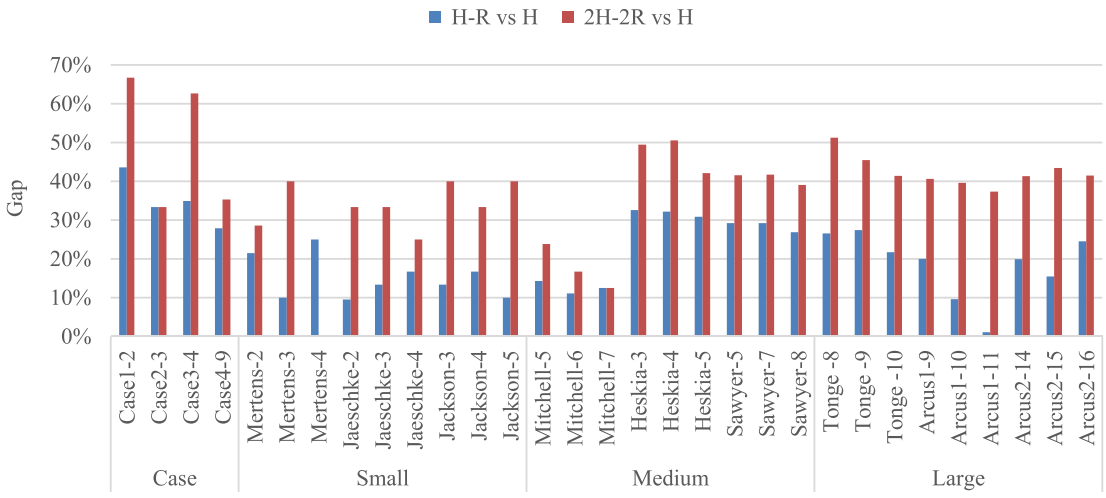


Fig. 12. Cycle time gaps for comparison of (1) H-R versus H and (2) 2H-2R vs. H.

7. Conclusions and future research directions

With the advent of recent advancements under the Industry 4.0 umbrella, increasing numbers of manufacturers have attempted to align with the emerging technologies developed in this era. In this regard, HRC is one of the technology enablers for transferring from machine-centered manufacturing towards a digital manufacturing environment. Inspired by a real-world case study from the automotive industry, this study addresses the ALBPs arising in HRC environments. Aside from the assignment of tasks to stations, the assignment of humans and robots to stations and the scheduling of tasks among them are performed while optimizing the cycle time and the total numbers of humans and robots as the primary and secondary objectives. In addition, multiple humans and robots at stations and joint tasks performed collaboratively by humans and robots are considered while taking both human and robot characteristics into consideration. To solve this problem (ALBP-HRC), a new MILP formulation is proposed. To solve medium to large-scale problems, a local search-based heuristic algorithm, namely SA with customized solution representations and neighborhood search operators, is developed to generate feasible solutions. The SA incorporates an adaptive mechanism in which the history of the algorithm is exploited to dynamically select the promising neighborhoods as the search evolves. The performance of the proposed MILP and SA are tested on real case studies and generated test problems. The computational study shows that the adaptive SA produces promising solutions relative to the MILP while outperforming other swarm intelligence algorithms, namely GA, PSO, and ABC, particularly in medium to large-sized problems. Furthermore, the analyses of human/robot versus HRC settings, particularly in the case studies, demonstrated that the cycle time could be significantly improved when multiple humans and robots with collaborative tasks are allowed at stations.

This study can be further extended by developing a cost-oriented formulation of the considered ALBP-HRC to include marginal expenses for humans and robots. Considering humans' well-being and postural risks while addressing the ALBP-HRC is another interesting topic. Other layouts of assembly lines such as U-shaped, parallel, and two-sided with HRC technology could be further investigated. Finally, novel metaheuristic algorithms for optimization of the extended ALBP-HRC can be additionally developed.

CRediT authorship contribution statement

Amir Nourmohammadi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Masood Fathi:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Amos H.C. Ng:** Validation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

The authors thank their industrial partners, Volvo Car Corporation and Jernbro AB, for their collaborative support during the project. The authors would also like to express their appreciation to anonymous referees for their useful remarks.

Funding

This study was initially supported by the European Union's Horizon 2020 research and innovation program under grant agreement, No. 723711, through the MANUWORK project. It was also partially funded by the Knowledge Foundation (KKS), Sweden, through the VF-KDO and ACCURATE 4.0 projects at the University of Skövde, Sweden, under grant agreements, No. 20180011 and No. 20200181, respectively.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cor.2021.105674.

References

Bruno, G., Antonelli, D., 2018. Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. Int. J. Adv. Manuf. Technol. 98, 2415–2427. https://doi.org/10.1007/s00170-018-2400-4.

Çil, Z.A., Li, Z., Mete, S., Özceylan, E., 2020. Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration. Appl. Soft Comput. 93, 106394 https://doi.org/10.1016/j.asoc.2020.106394.

Dalle Mura, M., Dini, G., 2019. Designing assembly lines with humans and collaborative robots: a genetic approach. CIRP Ann. 68, 1–4. https://doi.org/10.1016/j.cirp.2019.04.006.

De Nul, L., Breque, M., Petridis, A., 2021. Industry 5.0 - Towards a sustainable, human-centric and resilient European industry. 10.2777/308407.

Demir, K.A., Döven, G., Sezen, B., 2019. Industry 5.0 and Human-Robot Co-working. Procedia Comput. Sci. 158, 688–695. https://doi.org/10.1016/j.procs.2019.09.104.

Dianatfar, M., Latokartano, J., Lanz, M., 2019. Task balancing between human and robot in mid-heavy assembly tasks. Procedia CIRP 81, 157–161. https://doi.org/10.1016/j.procir.2019.03.028.

Ding, H., Schipper, M., Matthias, B., 2014. Optimized task distribution for industrial assembly in mixed human-robot environments - Case study on IO module assembly. In: 2014 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, pp. 19–24. https://doi.org/10.1109/CoASE.2014.6899298.

Fathi, M., Nourmohammadi, A., H.C. Ng, A., Syberfeldt, A., Eskandari, H., 2019. An improved genetic algorithm with variable neighborhood search to solve the assembly line balancing problem. Eng. Comput. 37, 501–521. doi:10.1108/EC-02-2019-0053.

Fletcher, S.R., Johnson, T., Adlon, T., Larreina, J., Casla, P., Parigot, L., Alfaro, P.J., Otero, M. del M., 2020. Adaptive automation assembly: identifying system requirements for technical efficiency and worker satisfaction. Comput. Ind. Eng. 139, 105772 https://doi.org/10.1016/j.cie.2019.03.036.

Gualtieri, L., Rauch, E., Vidoni, R., 2021. Methodology for the definition of the optimal assembly cycle and calculation of the optimized assembly cycle time in human-robot collaborative assembly. Int. J. Adv. Manuf. Technol. 113, 2369–2384. https://doi.org/10.1007/s00170-021-06653-y.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by Simulated Annealing. Science (80-.) 220, 671–680. https://doi.org/10.1126/science.220.4598.671.

Koltai, T., Dimény, I., Gallina, V., Gaal, A., Sepe, C., 2021. An analysis of task assignment and cycle times when robots are added to human-operated assembly lines, using mathematical programming models. Int. J. Prod. Econ. 242, 108292 https://doi.org/10.1016/j.ijpe.2021.108292.

Li, Z., Janardhanan, M.N., Ponnambalam, S.G., 2021a. Cost-oriented robotic assembly line balancing problem with setup times: multi-objective algorithms. J. Intell. Manuf. 32, 989–1007. https://doi.org/10.1007/s10845-020-01598-7.

Li, Z., Janardhanan, M.N., Tang, Q., 2021b. Multi-objective migrating bird optimization algorithm for cost-oriented assembly line balancing problem with collaborative robots. Neural Comput. Appl. https://doi.org/10.1007/s00521-020-05610-2.

Michalos, G., Spiliotopoulos, J., Makris, S., Chryssolouris, G., 2018. A method for planning human robot shared tasks. CIRP J. Manuf. Sci. Technol. 22, 76–90. https://doi.org/10.1016/j.cirpj.2018.05.003.

Petropoulos, D.I., Nearchou, A.C., 2011. A particle swarm optimization algorithm for balancing assembly lines. Assem. Autom. 31, 118–129. https://doi.org/10.1108/01445151111117700.

Raatz, A., Blankemeyer, S., Recker, T., Pischke, D., Nyhuis, P., 2020. Task scheduling method for HRC workplaces based on capabilities and execution time assumptions for robots. CIRP Ann. 69, 13–16. https://doi.org/10.1016/j.cirp.2020.04.030.

Taguchi, G., Chowdhury, S., Wu, Y., 2005. Taguchi's Quality Engineering Handbook. Wiley, Hoboken, NJ, USA.

Weckenborg, C., Spengler, T.S., 2019. Assembly Line Balancing with Collaborative Robots under consideration of Ergonomics: a cost-oriented approach. IFAC-PapersOnLine 52, 1860–1865. https://doi.org/10.1016/j.ifacol.2019.11.473.

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., Spengler, T.S., 2020. Balancing of assembly lines with collaborative robots. Bus. Res. 13, 93–132. https://doi.org/10.1007/s40685-019-0101-y.

Yaphiar, S., Nugraha, C., Ma'ruf, A., 2020. Mixed Model Assembly Line Balancing for Human-Robot Shared Tasks. pp. 245–252. doi:10.1007/978-981-15-0950-6_38.

Yu, V.F., Jewpanya, P., Redi, A.A.N.P., Tsao, Y.-C., 2021. Adaptive neighborhood simulated annealing for the heterogeneous fleet vehicle routing problem with multiple cross-docks. Comput. Oper. Res. 129, 105205 https://doi.org/10.1016/j.cor.2020.105205.