International Journal of Industrial Engineering Computations 13 (2022) 363–384

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Assembly line rebalancing and worker assignment considering ergonomic risks in an automotive parts manufacturing plant

Tolga Çimen[a*], Adil Baykasoğlu[a] and Sebnem Demirkol Akyol[a]

[a]*Dokuz Eylül University, Department of Industrial Engineering, İzmir, Turkey*

C H R O N I C L E

Article history:
Received November 21 2021
Received in Revised Format December 26 2021
Accepted January 27 2022
Available online January, 27 2022

Keywords:
Assembly line rebalancing problem
Worker assignment
Occupational repetitive action
Ergonomic risk assessment
Randomized constructive rule-based heuristic

A B S T R A C T

This paper recommends a new kind of assembly line rebalancing and worker assignment problem, taking ergonomic risks into account. Assembly line rebalancing problem (ALRBP) occurs when a current line must be rebalanced due to conditions such as changes in demand, production processes, product design, or quality issues. Although there are several research attempts on ALRBP in the relevant literature, only a few studies consider workers as unique individuals. This paper aims to solve the double reassignment problem: tasks to workers and workers to stations, considering ergonomic risk factors. This paper is the first study that comprises worker assignment and ergonomic constraints in ALRBP literature to the best of our knowledge. Objectives of our novel problem are to minimize rebalancing cost, which includes transportation of tasks and workers and minimize stations' ergonomic risk factors. A randomized constructive rule-based heuristic approach is developed to cope with the problem. The proposed solution approach is applied to benchmark data, and obtained results are promising. Moreover, the proposed solution approach is implemented in an automotive parts manufacturing plant.

1. Introduction

Assembly line balancing problem (ALBP) is relevant for optimizing one or more objectives and fulfilling task precedence relationships by assigning tasks, each with a processing time and a collection of precedence relationships, to stations (Özbakır & Seçme, 2020). To solve the ALBP, many algorithms and approaches have been applied since Salveson (1955) described the problem. Almost all of them have established an assembly line configuration from the beginning. However, in real life, a pre-balanced assembly line (AL) may have to be rebalanced for various reasons. The most usual reasons can be listed as; changes in cycle time due to demand fluctuations (Ağpak, 2010; Belassiria et al., 2018; Celik et al., 2014; Corominas et al., 2008; Mokhtari & Mozdgir, 2015; Serin et al., 2019), changes in product design and features (Altemeier et al., 2010; Fattahi & Samouei, 2016; Makssoud et al., 2013; Yang & Gao, 2016), changes in task times (Li & Boucher, 2017; Sikora et al., 2017; Zha & Yu, 2014), changes in task precedence relations (Gamberini et al., 2006; Zhang et al., 2018), and machine breakdowns (Ishigaki, 2018; Sancı & Azizoğlu, 2017). As a result, the assembly line rebalancing problem (ALRBP) occurs.

Many of the current AL rebalancing literature studies aim to rebalance the AL to reduce the number of changes to be made on the initial line. There is very limited work in the literature that tried to rebalance the AL, although the rebalancing problem typically occurs in real life very often. Moreover, worker assignment is barely studied in AL rebalancing literature. However, the assumption that all workers are identical does not represent the real-life case. Workers are unique in their ability, skill, and mentality. Hence, different workers operate the same task with different task times. Besides, in some cases, some workers could not operate specific tasks. Moreover, the mentioned rebalancing and worker assignment problems generally occur in manual ALs. Particularly in manual ALs, workers have certain occupational diseases due to repetitive unfavorable working conditions such as joint elbow syndromes and carpal tunnel. It takes a long time to treat certain occupational diseases, and

* Corresponding author Tel: +90 5053001886
E-mail: tolga.cimen@ogr.deu.edu.tr (T. Çimen)

2022 Growing Science Ltd.
doi: 10.5267/j.ijiec.2022.2.001

this causes a loss of productivity and workforce. Musculoskeletal conditions account for a significant portion of occupational illnesses in many European countries. In Turkey, approximately 40 percent of all occupational diseases are expected to be associated with musculoskeletal disorders (Berk, 2011). As a result of inadequate occupational ergonomics, work-related musculoskeletal problems arise. As a solution, workers' ergonomic stress should be held under a certain predetermined level due to exposure to repeated activities. This would eliminate occupational diseases and increase productivity. Although the influence of occupational exposure and ergonomics in production are becoming more and more important in practice, to our best knowledge, there has not been any research attempt that considers ergonomic aspects in the AL rebalancing literature. This study presents an assembly line rebalancing and worker assignment problem by considering ergonomic risk factors to narrow this distance between the literature and real-life (ErgoALWARBP). The presented ErgoALWARBP is interesting and novel since none of the previous work in the relevant literature handled the rebalancing of ALs under ergonomic and worker assignment constraints. The single assembly line balancing problem (SALBP) belongs to the NP-hard class of the combinatorial optimization problems (Karp, 1972). Since the ALRBP is a special case of the SALBP, it is NP-hard as well (Yang et al., 2013).

The remainder of this study is organized as follows. In Section 2, the literature is reviewed, and the justification of this research is stated. In Section 3, the novel ErgoALWARBP is detailedly defined in mathematical formulations and various additional constraints. In Section 4, the randomized constructive rule-based heuristic approach is introduced. In section 5, the proposed heuristic is tested using benchmark test problems, and the results of the experiments are discussed, followed by the application of the developed approach on a real-life case in Section 6, and section 7 concludes the article and suggests some future work.

2. Literature review

Gamberini et al. (2006) published the first study in the ALRBP literature. Authors considered task reassignment as an objective function and offered a single-pass heuristic with a multi-attribute decision-making system based on the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). Researchers focused on solving a single model stochastic rebalancing problem on straight lines. Due to the changes in cycle time and precedence relations, a rebalancing problem occurred. The authors stated an index to quantify the similarity of the task assignment between original and rebalanced lines, called the similarity factor. The objective was to minimize task reassignment by minimizing the similarity factor.

Several studies dealt with ALRBP for single-model, straight assembly lines, similar to the considered problem (Belassiria et al., 2018; Corominas et al., 2008; Fattahi & Samouei, 2016; Gamberini et al., 2006, 2009; Y. Li, 2017; Y. Li et al., 2021; Y. Li & Boucher, 2017; Makssoud et al., 2015, 2013; Mokhtari & Mozdgir, 2015; Rahman, 2010; Sancı & Azizoğlu, 2017). Gamberini et al. (2009) solved ALRBP with stochastic tasks with a heuristic algorithm. The objectives were to minimize task assignments' similarity and minimize expected assembly costs. Makssoud et al. (2015) suggested an exact solution method to solve the ALRBP of type-1. In their study, adding and removing tasks triggered the rebalancing, and the objectives were minimizing the number of stations and the number of modifications while considering negative task zoning constraints. Mokhtari and Mozdgir (2015) presented a differential evolution algorithm for solving ALRBP in which a new demand-related cycle time was needed. Researchers tried to optimize two objectives while rescheduling tasks to reduce cycle time: minimizing the incurred costs and non-smoothing the reconfigured line. Li and Boucher (2017) studied ALRBP of type-2, in which the assembly line was operated by robots, with stochastic task times. The authors offered a branch and bound method to deal with the rebalancing problem. Sancı and Azizoğlu (Sancı & Azizoğlu, 2017) stated a GA which was hybridized with a priority rule-based heuristic procedure for solving ALRBP. The objectives in their study were maximizing line efficiency and workload balance while minimizing total cost. Belassiria et al. (2018) solved single-model ALRBP with a hybrid genetic algorithm. The authors used prioritized rules to maximize the line's efficiency and minimize the rebalancing cost concurrently. Li et al. (2021) studied the AL rebalancing problem due to sudden disruption. They considered stochastic operation times, and the goal was minimizing total cost. The authors compared two solution approaches: i) periodic and ii) data-driven rebalancing policies. The results show that a data-driven rebalancing policy is better than a periodic rebalancing policy.

Few studies deal with the single model U-shaped ALRBP (Ağpak, 2010; Celik et al., 2014; Serin et al., 2019; Sikora et al., 2017; Zha & Yu, 2014). Workers can work on one of each sub-line or both of them simultaneously in U-shaped assembly lines (Ramezanian & Khalesi, 2021). Ağpak (2010) focused on rebalancing U-shaped and straight assembly lines with a heuristic method while considering task precedence constraints. Zha and Yu (2014) presented a hybrid ant colony optimization and filtered beam search for solving ALRBP. The authors considered multiple objectives: minimizing the number of stations, machine costs, labor costs, and walking time of operators. Unlike previous U-shaped studies, Celik et al. (2014) studied ALRBP with stochastic task times. To reduce the overall cost of rebalancing, the authors used an ant colony optimization algorithm. The rebalancing cost consists of task transportation costs, station opening/closing costs, and station operating costs while considering the probability of incompletion cost. Serin et al. (2019) stated a GA for solving U-type ALRBP with stochastic task times. Their objectives were to minimize the number of stations and total rebalancing costs.

A two-sided assembly line is another widely used line type that there are two stations on two sides of a conveyor belt. The number of studies on two-sided ALRBP is limited ((Grangeon et al., 2011; Liu et al., 2012; Y.-H. Y. Zhang et al., 2018; Y. Zhang et al., 2018, 2020)) Liu et al. (2012) studied the rebalancing problem focused on two-side assembly lines. The authors studied the problem of how to reconfigure the conveyor assembly line to serus. Seru is a Japanese word for cell and usually refers to assembly cell. The authors gave a mathematical model that addressed two issues: the number of serus that should be established and the number of workers that should be assigned to each seru. They stated that the computational results were promising. Zhang et al. (2018) modified non-dominated sorting GA for solving a real-life multi-objective ALRBP of type-2 on two-sided lines. The objectives of their problem were minimizing cycle time while simultaneously minimizing rebalancing cost and minimizing similarity.

Several product versions are manufactured simultaneously in some instances, which is called mixed-model (Mardani-Fard et al., 2020) ALRBP. Altemeier et al. (2010) defined a mathematical optimization method for solving mixed-model ALRBP. They considered task precedence restrictions, workplace limits (average workload on a station), and fixed processes (processes that could not be moved to other stations) as validity constraints while minimizing rebalancing costs. Grangeon et al. (2011) presented three heuristics for solving mixed-model two-sided ALRBP. The authors tried to minimize task relocation and workload smoothing. Ishigaki (2018) addressed a mixed-model ALRBP with station-dependent processing times. A heuristic method was developed to minimize the line stopping time.

Several studies dealt with AL balancing and worker assignment. Even though they did not consider the rebalancing process, they gave initiation for rebalancing and worker assignment simultaneously (Borba & Ritt, 2014; Chaves et al., 2007; Costa & Miralles, 2009; Guo et al., 2008; Miralles et al., 2008; Moreira et al., 2015; Zacharia & Nearchou, 2016). However, published literature on ALRBPs often only focuses on the rebalancing of the assembly line. There is limited work that considers worker assignment together with the rebalancing problem (Corominas et al., 2008; Fattahi & Samouei, 2016; Girit & Azizoğlu, 2021; Liu et al., 2012; Sikora et al., 2017; Yang et al., 2013; Yang & Gao, 2016). Corominas et al. (2008) solved ALRBP in a motorcycle plant with a linear programming procedure while minimizing the number of temporary workers. Temporary workers needed more time to implement the tasks and could not perform some of them. Yang et al. (Yang et al., 2013) solved mixed-model ALRBP with a multi-objective genetic algorithm while minimizing workload smoothness, rebalancing cost, and the number of stations. Fattahi and Samouei (2016) developed a heuristic-simulation algorithm for mixed-model ALRBP of type-2 with worker assignment. The objective was to minimize cycle time with stochastic task times without changing the number of stations while considering task precedence and worker skill constraints. Yang and Gao (2016) offered a branch, bound, and remember algorithm for solving mixed-model ALRBP of type-1. A model for adjacent workforce cross-training policy assembly lines was presented, in which two employees in neighboring stations could learn the skills of each other. When there was a change in product demand or product mix, the tasks could only be shifted to adjacent stations to rebalance the line, where there was a worker who had previously learned the skills to process the task. Sikora et al. (2017) studied the U-shaped ALRBP of type-2 with worker assignment. The authors tried to minimize cycle time and total movement time. Simultaneously, traveling worker ALBP was solved, enabling workers to execute tasks in multiple stations. The authors stated a mixed-integer linear programming method which includes zoning and capacity constraints, human and robotic worker assignment restrictions, and set-up times. Girit and Azizoğlu (2021) rebalanced AL with two objectives: i) maximizing workload balance and ii) minimizing total replacement distance. They offered a fairness index to penalize higher workloads and smooth the work among stations. However, they assumed that all workers were identical.

Note that none of the above-mentioned work considered workers as a unique resource. Although they considered the workforce while rebalancing, none of them considered varying operation times of tasks due to workers. Since the motivation of the worker assignment problem introduced by Miralles et al. (2008) is to address every worker as a unique resource, in this study, each task has a specific processing time depending on the worker. In the AL literature, labor uniqueness is typically underrated. However, the levels of skill, ability, experience, and physical conditions of workers directly affect the operation times of tasks. Limited work considers the importance of unique worker skills in the manufacturing environment (Fichera et al., 2017; Tóth & Kulcsár, 2021). Fichera et al. (2017) tackled the worker assignment problem in a cellular manufacturing plant by considering the different ability levels of workers and the learning effect. Tóth and Kulcsár (2021) studied worker-dependent processing times in flexible manufacturing systems. The authors proposed a solution procedure by using simulation and demonstrated the effect of heterogeneous workers. Although these papers pay regard to unique workers and varying operation times of tasks, they study manufacturing environments rather than ALs. It is understood from the literature survey that there is a gap in the relevant literature that none of the rebalancing papers considers varying operation times of tasks depending on unique workers, which is one of the essential contributions of this paper.

Moreover, note that none of the above-mentioned research studies consider ergonomic risk factors of the working environment. The limited number of ALRBP studies with worker assignments mentioned above do not consider ergonomic issues as a constraint or in the objective function. Ergonomic aspects are becoming increasingly crucial in real-life scenarios, especially in manual assembly lines. Manual assembly lines cover a major part of manufacturing in Turkey. So, this is the main motivation of this research. This research study is a pioneer because while tackling ergonomic risks of the manufacturing environment, the presented method reassigns workers to stations and reassigns tasks to workers simultaneously. The motivation of this work is to fill the gap in the relevant literature.

3. Assembly Line Rebalancing and Worker Assignment Problem Considering Ergonomic Risk Factors (ErgoALWARBP)

As stated in the previous section, ALRBPs have barely been studied compared to ALBPs. Also, most of the rebalancing studies accept that all workers are identical in the relevant literature. However, this assumption is not realistic because workers have unique characteristics (Miralles et al., 2008). Especially in manual ALs, workers must be considered unique to represent the real-life manufacturing environment better. Moreover, it is essential to assign tasks as equally as possible among workers when rebalancing an existing line. Such assignment enables smoothening the strain levels of workers among stations (Otto &

Scholl, 2011). For ergonomic risk assessment in ALs, many methods have been used, such as Occupational Repetitive Action (OCRA)(Occhipinti, 1998), The European Assembly Worksheet (EAWS) (Schaub et al., 2013), the National Institute for Occupational Safety and Health (NIOSH) (Waters, 1994), Quick Exposure Check (QEC)(G. Li & Buckle, 1999), Rapid Upper Limb Assessment (RULA)(McAtamney & Nigel Corlett, 1993), and Rapid Entire Body Assessment (REBA)(Hignett & McAtamney, 2000). The OCRA method is used for estimating risk factors for tasks done by upper limbs repetitively with high frequency. In manual assembly lines, especially in automotive harness production, workers use their upper limbs repetitively in high frequency. Therefore, the OCRA method is used for ergonomic risk calculations in this study. Further details about ergonomic risk assessment and the OCRA method are described by Akyol and Baykasoğlu (Akyol & Baykasoğlu, 2019a).

3.1 Assumptions

The following assumptions are taken into account for the problem:

- It is considered that: the line was previously balanced before the disruptive event occurred.
- Each product model's operating times and precedence diagrams are given.
- A station can perform any operation, and a task can be allocated to one station.
- Workers' travel times are neglected.
- The number of workers available for reassignment is greater than the number of workers in the initial assignment to fit the worker-need in case of opening additional stations while rebalancing the AL.

3.2 Notations

The notation used in the rest of the paper is as follows (Celik et al., 2014; Gamberini et al., 2006; Miralles et al., 2008; Yang et al., 2013):

$i.j$	task index, for $i.j = 1.\dots.n$
$a.b$	station index, for $a.b \in S$
$\boldsymbol{TC_{rb}}$	**the total cost of rebalancing**
$cost_{open}$	opening costs of a new station
$cost_{close}$	cost of closing an existing station
$cost_{run}$	operating cost of a station for a month
$cost_i$	transportation cost of task *i*
N_{rb}	set of tasks that were transported for rebalancing
m	number of stations
m^0	initial station count
m^+	number of new stations that were opened after rebalancing
m^-	number of stations that were closed after rebalancing
$\boldsymbol{MSF}$	**mean similarity factor of the line**
SF_i	similarity factor of task *i*, for $i \in N$
TIB_i	set of tasks allocated to the same station as task *i* in the initial balancing, other than task *i*, for $i \in N$
TNB_i	set of tasks allocated to the same station as task *i* in the new balancing, other than task *i*, for $i \in N$
n	number of tasks
n^0	number of tasks in the initial balancing
$\boldsymbol{WorkerMSF}$	**mean similarity factor of the workers**
$WorkerSF_W$	similarity factor of worker *w*, for $w \in W$
w	worker index, for $w \in W$
W	set of workers
N_W	set of tasks assigned to worker *w*, for $w \in W$
N_W^0	initial set of tasks assigned to worker *w,* for $w \in W$
$\boldsymbol{madOCRA}$	**mean absolute deviation of OCRA index values**
$avgOCRA$	OCRA average
$OCRA_a$	OCRA index value of station a
d_i^-	negative deviation
d_i^+	positive deviation
$\boldsymbol{n_{rb}}$	**number of tasks that were transported for rebalancing**
NS_a^0	initial set of tasks assigned to station *a*, for $a = 1.\dots.m^0$
NS_a	set of tasks assigned to station *a*, for $a = 1.\dots.m$

LE	**line efficiency**
SX	**similarity index**
c	cycle time of the line
Tt_a	total task processing time of station a (station time)
S	set of stations
N	set of tasks
y_{aw}	binary variable equal to 1, only if worker *w* is assigned to station *a*
x_{iaw}	binary variable equal to 1, only if task *i* and worker *w* are assigned to station *a*
t_{iw}	processing time of task *i* when worker *w* executes it
IP_j	immediate predecessors of task *j*

3.3 Objective functions

It is benefited from several previous studies in this study while defining the seven objective functions and their formulations. (Celik et al., 2014; Gamberini et al., 2006; Yang et al., 2013; Zacharia & Nearchou, 2016). The objectives are listed below:

$$\boldsymbol{min}.TC_{rb} = (cost_{open} * m^{+}) + (cost_{close} * m^{-}) + (cost_{run} * (m - m^{0})) + \sum_{i \in N_{rb}} cost_i \tag{1}$$

$$m^{+} = m - m^{0}, \qquad m^{-} = 0;\ if\ m > m^{0} \tag{1.1}$$

$$m^{-} = m^{0} - m, \qquad m^{+} = 0;\ if\ m < m^{0} \tag{1.2}$$

$$\boldsymbol{max}.MSF = \left(\sum_{i=1}^{n} SF_i\right)/n \tag{2}$$

$$SF_i = \frac{Cardinality\left(TIB_i \cap TNB_i\right)}{Cardinality\left(TIB_i\right)} \tag{2.1}$$

$$TIB_i = \{\, j \in NS_a^0 \mid i \in NS_a^0\ and\ j \neq i \,\}, \qquad for\ i = 1, \dots, n^0 \tag{2.2}$$

$$TNB_i = \{\, j \in NS_a \mid i \in NS_a\ and\ j \neq i \,\}, \qquad for\ i = 1, \dots, n \tag{2.3}$$

$$\boldsymbol{max}.WorkerMSF = \left(\sum_{w \in W} WorkerSF_W\right)/m \tag{3}$$

$$WorkerSF_W = \frac{Cardinality(N_W^0 \cap N_W)}{Cardinality(N_W^0)}, \qquad \forall w \in W \tag{3.1}$$

$$\boldsymbol{min}.madOCRA = \left(\sum_{a=1}^{m} |avgOCRA - OCRA_a|\right)/m \tag{4}$$

$$avgOCRA = \left(\sum_{a=1}^{m} OCRA_a\right)/m \tag{4.1}$$

$$\boldsymbol{min}.n_{rb} = \sum_{a=1}^{m}\left(Cardinality(NS_a^0) - Cardinality(NS_a^0 \cap NS_a)\right) \tag{5}$$

$$\boldsymbol{max}.LE = \left(\sum_{a=1}^{m} Tt_a\right)\Big/(m * c) \tag{6}$$

$$\boldsymbol{min}.SX = \sqrt{\sum_{a=1}^{m} (c - Tt_a)^2} \tag{7}$$

As Celik et al. (2014) stated, some tasks may be transferred between stations to rebalance the assembly line, and a new line balance may need more or a smaller number of stations. In the current study, the first objective (Eq. (1)) is minimizing total rebalancing cost (TC_{rb}) which is calculated as stated in Eqs. (1), (1.1), and (1.2) (Celik et al., 2014).

- TC_{rb} is calculated by summing up station opening/closing/operating and task transportation costs.

- $cost_i$ arise if certain tasks are assigned from their current positions to new positions (Celik et al., 2014).
- $cost_{open}$ happens due to the recruiting of new workers and the procurement of new equipment as the number of stations rises in the new line balance (Eq. (1.1)).
- $cost_{close}$ occurs because of equipment disposal and the dismissal of obsolete staff when the number of stations in the new line balance decreases (Eq. (1.2)).
- $cost_{run}$ is calculated by dividing the sum of operating expenses such as rent, equipment, labor, inventory costs by a period (days, months, production hours). Therefore, the value changes (the duration that the new line balance will operate. If $m^+ > 0$ then the total operating cost is positive, and if $m^- > 0$, the total operating cost is negative.

Gamberini et al. (2006, 2009) presented an objective called mean similarity factor (MSF), which is the similarity between the rebalanced AL and the initial AL. The second objective (Eq. (2)) in the current study is to minimize the MSF. To evaluate MSF, TIB_i, and TNB_i are introduced in Eqs. (2.2) and (2.3) respectively. SF_i, which is calculated using Eq. (2.1), that is obtained by dividing the number of tasks assigned to the same station as $task\ i$ in the initial and new AL, by to the number of tasks assigned to the same station in the initial AL. The third objective (Eq. (3)) is stated for the first time in the literature to maximize the similarity of the tasks performed by workers between the initial and new AL (*WorkerMSF*), which evaluates the degree of similarity between the initial and the new task assignment of each worker. The similarity factor of the generic $worker\ w$ ($WorkerSF_W$), which is calculated using Eq. (3.1), is obtained by dividing the number of tasks assigned to $worker\ w$ both in the initial and in the new AL, by the number of tasks assigned to the $worker\ w$ in the initial AL. For example, let us assume that $worker$ 1 is performing the tasks {1, 2, 3, 4} in $station$ 1 in the initial AL. After the rebalancing, $worker$ 1 is assigned to station 2 and starts performing the tasks {3, 4, 5, 6}. The tasks {3, 4} are performed by $worker$ 1 in both the initial and the new AL, therefore *WorkerSF₁* is equal to *Cardinality(3,4) / Cardinality(1,2,3,4) = 2 / 4 = 0,5.* After applying this procedure to all assigned workers, $WorkerMSF$ value is calculated by summing up the $WorkerSF$ values divided by the number of stations in the rebalanced AL as stated in Eq. (3). In this study, ergonomic risk levels of stations are determined by using OCRA (Occhipinti, 1998) ergonomic risk assessment method (for more information, see Appendix; Akyol and Baykasoğlu, (Akyol & Baykasoğlu, 2019a). The fourth objective (Eq. (4)) is minimizing the mean absolute deviation of stations' OCRA index values (*madOCRA*), which was offered to control and monitor the ergonomic risk level of the stations and to smooth the workload between workers in terms of ergonomic risks. To calculate *madOCRA,* the average of OCRA index values (*avgOCRA*) is needed, which is evaluated using Eq. (4.1). Task transportation cost is an essential part of the total rebalancing cost. Therefore, we also try to minimize the number of tasks that are transported for rebalancing, n_{rb}, which is our fifth objective (Eq. (5)). Although two solutions can have the same MSF value, they can have a different number of reassigned tasks. The sixth objective (Eq. (6)) is maximizing *LE*, and the seventh objective (Eq. (7)) is minimizing the *SX* (Zacharia & Nearchou, 2016), which is a measure to distribute the work equally between the stations.

3.4 Constraints

Constraints of the mathematical model for the ALRBP with worker assignment can be stated as follows (Gamberini et al., 2006; Miralles et al., 2008):

$$\sum_{w \in W} \sum_{a \in S} x_{iaw} = 1, \qquad \forall i \in N, \tag{8}$$

$$\sum_{a \in S} y_{aw} \leq 1, \qquad \forall w \in W, \tag{9}$$

$$\sum_{w \in W} y_{aw} \leq 1, \qquad \forall a \in S, \tag{10}$$

$$\sum_{w \in W} \sum_{a \in S} (s * x_{iaw}) \leq \sum_{w \in W} \sum_{a \in S} \left(s * x_{jaw}\right), \quad \forall i, j / i \in IP_j, \tag{11}$$

$$\sum_{i \in N} \left(t_{iw} * x_{iaw}\right) \leq c, \qquad \forall w \in W, \qquad \forall a \in S, \tag{12}$$

$$\sum_{i \in N} x_{iaw} \leq (m * y_{aw}), \qquad \forall w \in W, \qquad \forall a \in S, \tag{13}$$

with

$y_{aw} \in \{0, 1\}, \qquad \forall a \in S, \qquad w \in W,$

$x_{iaw}, x_{jaw} \in \{0, 1\}, \qquad \forall a \in S, \qquad w \in W, \qquad i, j \in N$

Constraint (8) ensures that each $task\ i$ is assigned to a single $worker\ w$ and a single $station\ a$. Constraint sets (9) and (10) express that a worker can work at only one station, and there must only be one assigned worker in each station, respectively. The precedence relationships between task *i* and task *j* are specified in constraints set (11), in which task *i* is the predecessor of task *j*. Constraint sets (12) and (13) guarantee that when cycle time *c* is not reached, any worker *w* assigned to the station *a* can process more than one task. Nevertheless, the proposed problem is an extension of the ALRBP with worker assignment because it includes ergonomic risk assessment. So, the OCRA index value ($OCRA_a$) calculations can be included in the model implicitly. Equation (14) demonstrates that risk analysis for all the stations is smoothed in accordance with the average OCRA index value ($avgOCRA$).

$$OCRA_a + d_i^+ - d_i^- = avgOCRA \tag{14}$$

Note that the above constraints are given to define the proposed problem formally. Because of the complexity, the novel ErgoALWARBP cannot be solved by exact solution methods. A randomized constructive rule-based heuristic approach is developed to tackle the problem.

4. The Proposed Algorithm for ErgoALWARBP

In this study, task and worker selection rules which were described by Akyol and Baykasoğlu (2019b), are applied to pick appropriate tasks for workers. A randomized constructive rule-based heuristic approach is developed to find a feasible solution. Firstly, 39 tasks and four worker priority rules are set. Then, the heuristic prioritizes the tasks, and a task is chosen via roulette wheel selection. The heuristic reassigns tasks sequentially to the stations. Later, after applying worker selection rules, a worker among prioritized workers is selected using roulette wheel selection and reassigned to that station. When cycle time is exceeded, a new station is opened. In this way, the developed heuristic constructs a new solution step by step and includes rule-based prioritized randomness (for more information, see Akyol and Baykasoğlu, (2019b)). After obtaining a newly rebalanced AL, the new line is compared with the old one regarding the MSF of tasks and workers.

4.1. The rule-based task selection algorithm

In this study, 39 task priority rules (Akyol & Baykasoğlu, 2019b) are applied, and assignable tasks are prioritized by considering different task execution times depending on the operators. First, on lines 5 to 7, one point is added to each task that can be assigned to the station. Then, on lines 9 to 11, each task selection rule is called one by one. The rules loop list of assignable tasks and add point(s) to the appropriate one(s). These task selection rules are detailly described in Akyol and Baykasoğlu (2019b). Then, a task is chosen among assignable tasks such that the task with the higher point has a higher probability of selection on line 12. In the literature, this method is called *roulette wheel selection*. On line 13, the selected task by roulette wheel selection is returned. If there is only one assignable task to the station, the rules are not applied, and the task is directly returned on line 15. The overall procedure of the rule-based task selection method is outlined in Algorithm 1 as follows:

Algorithm 1: The rule-based task selection algorithm

```
1    METHOD: Rule-based task selection for the station
2    INPUT: List of assignable tasks, workers with task processing times
3    OUTPUT: Selected task
4
5    FOR each task in the station's assignable tasks list DO
6      Add 1 point to the task
7    END FOR
8    IF station's assignable tasks count > 1 THEN
9      FOR each rule in 39 task rules DO
10       CALL rule //add point(s) to the appropriate task(s)
11     END FOR
12     SelectedTask ← CALL Task_Roulette_Wheel_Selection (list of assignable tasks)
13     RETURN SelectedTask
14   ELSE IF assignable tasks count = 1 THEN
15     RETURN the only assignable task
16   END IF
```

4.2. The rule-based worker selection algorithm

In this study, four rules for the selection of workers are applied (Akyol & Baykasoğlu, 2019b):

- RND - Random priority
- GNTE - Greatest no. of tasks executed
- GNTEMT - Greatest no. of tasks executed in minimum time
- MU - Maximum utilization

First, on lines 5 to 7, one point is added to each worker that can be assigned to the station. If there is more than one assignable worker, then these four rules are called on lines 9 to 16. Each rule adds two points to the appropriate worker. By utilizing the above four rules, assignable workers are prioritized. Then, by applying the *roulette wheel selection* described in section 4.1, a worker is chosen among assignable workers. On line 18, the selected worker by roulette wheel selection is returned. If there is only one assignable worker to the station, the rules are not applied, and the worker is directly returned on line 20. The rule-based worker selection method is outlined in Algorithm 2 as follows:

Algorithm 2: The rule-based worker selection algorithm

```
1   METHOD: Rule-based worker selection for the station
2   INPUT:  Station's assigned tasks, list of assignable workers with task processing times
3   OUTPUT: Selected worker for the station
4
5   FOR each worker in the list of assignable workers DO
6     Add 1 point to the worker
7   END FOR
8   IF assignable workers count > 1 THEN
9     //rule-1 (RND)
10    Add 2 points to the randomly selected worker from the assignable workers list
11    //rule-2 (GNTE)
12    Add 2 points to the worker who is capable of processing max. number of tasks
13    //rule-3 (GNTEMT)
14    Add 2 points to the worker who can process max. the number of tasks in min. time
15    //rule-4 (MU)
16    Add 2 points to the worker who can execute this station's tasks fastest
17    SelectedWorker ← CALL Worker_Roulette_Wheel_Selection (list of assignable workers)
18    RETURN SelectedWorker
19  ELSEIF assignable workers count = 1
20    RETURN the only assignable worker
21  END IF
```

4.3. The developed preemptive goal programming algorithm for fitness evaluation

There are several ways for evaluation of solutions, such as comparing cycle time (Fattahi & Samouei, 2016; Li, 2017; Liu et al., 2012; Sancı & Azizoğlu, 2017; Sikora et al., 2017; Zhang et al., 2018; Y. Zhang et al., 2018, 2020), total cost (Lai et al., 2015; Serin et al., 2019; Zha & Yu, 2014), line efficiency (Altemeier et al., 2010; Belassiria et al., 2018; Rahman, 2010), workload balance (Grangeon et al., 2011; Lai et al., 2015; Mokhtari & Mozdgir, 2015; Oliveira et al., 2012) or similarity index (Zacharia & Nearchou, 2016).

Algorithm 3: The preemptive goal programming algorithm (for Goal 1)

```
METHOD: Return best solution with goal programming
INPUT:  Feasible solutions, Goal's objective order
OUTPUT: The best solution

INIT BestSolution
FOR each solution in feasible solutions DO
  //obj.1 Total cost
  IF solution's Total Cost < BestSolution's Total Cost THEN
    BestSolution ← solution
  ELSE IF solution's Total Cost = BestSolution's Total Cost THEN
    //obj.2 Mean similarity factor (MSF)
    IF solution's MSF > BestSolution's MSF THEN
      BestSolution ← solution
    ELSE IF solution's MSF = BestSolution's MSF THEN
      //obj.3 MSF Worker
      IF solution's WorkerMSF > BestSolution's WorkerMSF THEN
        BestSolution ← solution
      ELSE IF solution's WorkerMSF = BestSolution's WorkerMSF THEN
        //obj.4 Mean absolute deviation of OCRA index values
        IF solution's madOCRA < BestSolution's madOCRA THEN
          BestSolution ← solution
        ELSE IF solution's madOCRA = BestSolution's madOCRA THEN
          //obj.5 Re-assigned tasks count
          IF solution's MovedTasksCount < BestSolution's MovedTasksCount THEN
            BestSolution ← solution
          ELSE IF solution's MovedTasksCount = BestSolution's MovedTasksCount THEN
```

```
27                     //obj.6 Line efficiency
28                     IF solution's LE > BestSolution's LE THEN
29                       BestSolution ← solution
30                     ELSE IF solution's LE = BestSolution's LE THEN
31                       //obj.7 Line smoothness index
32                       IF solution's SX < BestSolution's SX THEN
33                         BestSolution ← solution
34                       END IF //obj.7
35                     END IF //obj.6
36                   END IF //obj.5
37                 END IF //obj.4
38               END IF //obj.3
39             END IF //obj.2
40           END IF //obj.1
41         END FOR
42         RETURN BestSolution
```

We made preferences among the seven objectives and put them in an initial order (Goal 1) as in Algorithm 3. The preemptive goal programming algorithm for Goal 1 is outlined in Algorithm 3, that the highest priority is finding the solution with the minimum total rebalancing cost. Firstly, we initialize the *BestSolution* variable on line 5. For the objectives that are trying to be minimized, the properties in the *BestSolution* are initialized with a very high number H. Likewise, for the objectives that are trying to be maximized, the properties in the *BestSolution* are initialized with a very low number L. By this way, the first feasible solution that is compared with the *BestSolution* is always better and is assigned to the *BestSolution*. By looping all feasible solutions on line 6, we compare solution's total cost, MSF, MSF Worker, mean absolute deviation of OCRA index values, line efficiency, and line smoothness index values with the *BestSolution*'s appropariate values respectively on lines 7 to 33. When the compared property of the solution is better than *BestSolution*'s value, the solution is assigned to the BestSolution. The calculation details of the relevant objectives were shared in section 3.3.

The priority of the objectives may change according to current conditions. To fit those varying needs, the initial order of the objectives, which is defined in Algorithm 3, was changed using insert, remove, and shift operations. In this way, six more objective orders (goals) were generated inside our solution calculation method as outlined in Algorithm 4.

Algorithm 4: Calculating the best solution for all goals (changing order of objective functions)

```
METHOD: Calculate goal results
INPUT:  Initial order of objective functions (list of obj. functions)
OUTPUT: The best solution for each goal (list of best solutions)

BestSolution ← CALL Algorithm_3 (Initial order of obj. functions, list of feasible solutions)
Insert BestSolution into the list of best solutions

FOR each objective in the list of obj. functions DO
  firstObjective ← objective
  Remove objective from the list of obj. functions
  Shift the objectives before the removed objective to right
  Insert firstObjective to the first position of the list of obj. functions

  BestSolution ← CALL Algorithm_3(list of obj. functions, list of feasible solutions)
  Insert BestSolution into the list of best solutions
END FOR
RETURN the list of best solutions
```

The objective order of the seven goals are listed below:

G1 – min. TC_{rb}:	$TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > LE > SX$
G2 – max. MSF:	$MSF > TC_{rb} > WorkerMSF > madOCRA > n_{rb} > LE > SX$
G3 – max. WorkerMSF:	$WorkerMSF > TC_{rb} > MSF > madOCRA > n_{rb} > LE > SX$
G4 – min. madOCRA:	$madOCRA > TC_{rb} > MSF > WorkerMSF > n_{rb} > LE > SX$
G5 – min. n_{rb}:	$n_{rb} > TC_{rb} > MSF > WorkerMSF > madOCRA > LE > SX$
G6 – max. LE:	$LE > TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > SX$
G7 – min. SX:	$SX > TC_{rb} > MSF > WorkerMSF > madOCRA > n_{rb} > LE$

The rule-based task and worker selection algorithms were described with an illustrative example in Akyol and Baykasoğlu (2019b). The proposed heuristic solution procedure for ErgoALWARBP is described with the flowchart that is shared in Fig.1.

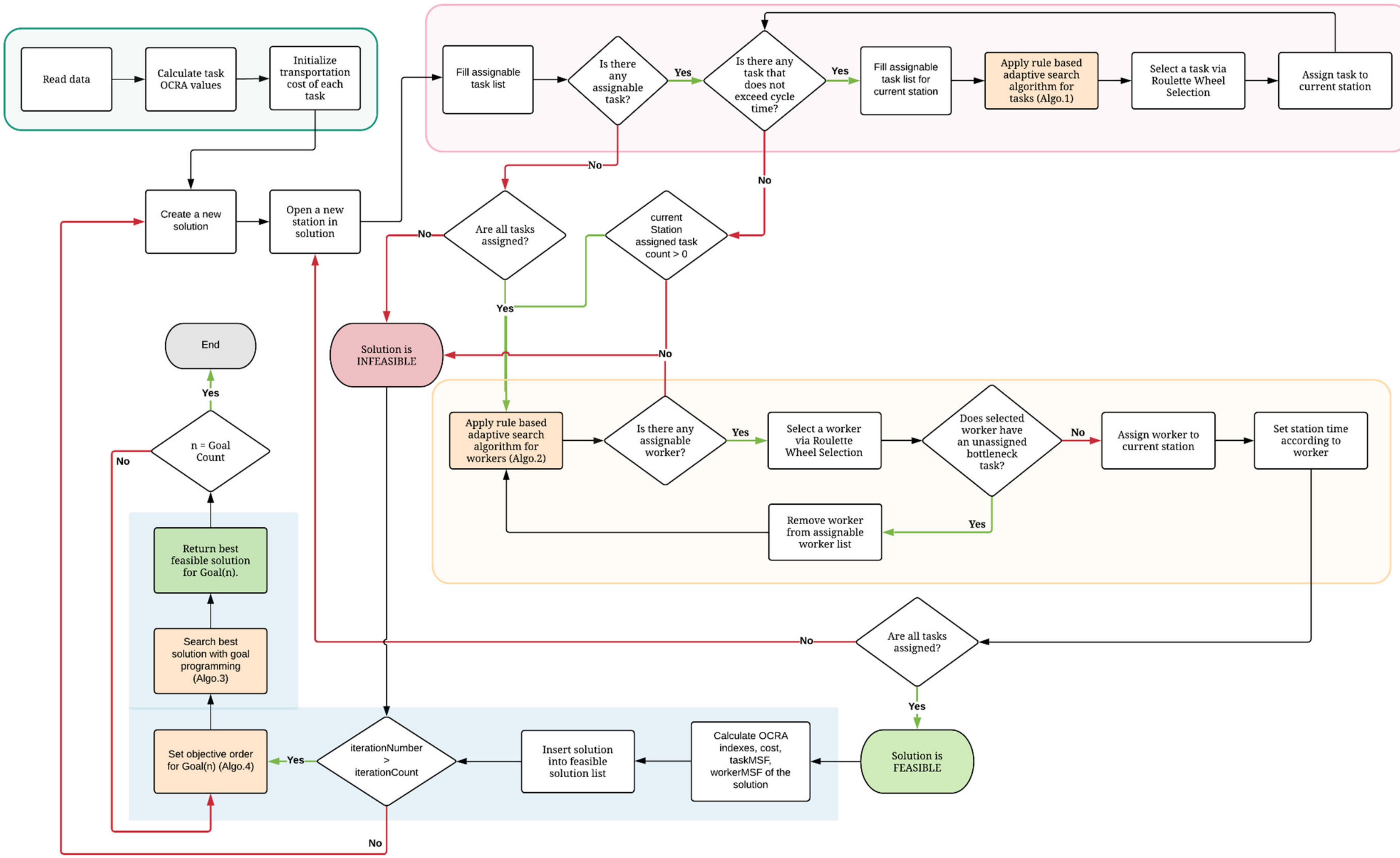

Fig.1 Flowchart of the proposed heuristic solution procedure

5. Computational Study

We evaluated the rebalancing performance of our proposed heuristic with well-known test problems of Scholl (1993, 1999). Our problem set contains 25 test problems in which the number of tasks is between 7 and 297, and cycle times are between 7 and 5853 seconds. In Table 1, the benchmark data family of the precedence graph, number of tasks, the cycle time of the AL before rebalancing (**initial cycle time**), and the minimal number of stations for the new cycle time (**m***) are shown in the first four columns, respectively. The initial balance is generated using the COMSOAL (Computer Method of Sequencing Operations of Assembly Lines) heuristic that was developed by Arcus (1966). For each test problem, 500 feasible solutions (12500 solutions in total) are calculated using COMSOAL that randomly assigns tasks to stations considering precedence relations. Then, the solution with minimum cycle time and maximum line efficiency is returned as the initial solution for each problem. Changes in customer demand led to an increase (or decrease) in the initial cycle time, which caused the rebalancing need. The updated cycle time values used for rebalancing are shared in the "New cycle time" column in Table 1. Each data family's new cycle time values are randomly chosen from the benchmark dataset. In this way, the reassignment results can be compared with **m***, which was calculated with exact algorithms and shared in the benchmark dataset. The developed heuristic runs for 1000 iterations for each test problem (25000 iterations in total).

5.1. Conputational test results

The best and average results found for each test problem are shared in Table 1. Although the suggested heuristic solution method calculates results for seven different objectives, the benchmark problems were only solved to minimize the number of stations. Therefore, we can only measure the performance of the results in terms of the number of stations, but the result set also contains CPU time, smoothness index, line efficiency, MSF, and the number of tasks moved for informational purposes.A rebalanced solution with a line efficiency that is greater than 99% is found for Barthold, Gunther, Heskiaoff, and Mansoor. 34.5% of the tasks are moved on average to find a rebalanced solution with a minimum number of task moves for each test problem. The proposed heuristic found the minimum number of stations in 84% (21 of 25) of the test problems. For Barthol2, Lutz2, Scholl, and Warnecke, which can be defined as large-size problems with the number of tasks 148, 89, 297, and 58 respectively, a solution with one more station than the optimum solution is found. In comparison with our 25 test problems from Sholl's dataset (1993, 1999), Belassiria et al. (2018) used and solved nine of the test problems. Both algorithms reached the minimal number of stations **m*** for Hahn, Heskiaoff, Lutz1, and Mukherje, but our heuristic method found a better solution (fewer stations) than the hybrid genetic algorithm of Belassiria et al. (2018) for Barthold, Gunther, Lutz2, Scholl, and Warnecke.

6. Industrial case study

A real-world case study with a sample of actual operational data is implemented to show the effectiveness of the suggested method. The company has 126 locations worldwide, producing several different products for the automotive sector. The industrial case is presented at a wiring harness production plant of the company in Turkey, which manufactures cable networks for the most prominent automotive brands in the world. Several components are required to assemble automotive wiring harnesses, such as wires, connectors, tubes, terminals, tapes, etc. As customer needs differ widely, more than four thousand components are used for harness production in the current plant.

6.1 Operational problem information

The manufacturing consists of four main processes: cutting, sub-assembly, main assembly, and electrical testing. In this study, the main assembly part is considered for ALRBP. The car harnesses are divided into smaller parts such as engine bay (EB), floor, roof, doors, and interior panel (IP) to ease the harness's production, logistics, and installation into the car. A harness is assembled in the main assembly area on vertically positioned boards which are specially designed and produced for each harness. These production boards are named *stations,* and AL contains m identical stations. The studied assembly line is O-shaped, and stations are moving with a constant speed that finishes its tour in C seconds. A worker is assigned to each station and positioned outside of the O-shape line and only works in the assigned station. This is contrary to U-shaped lines where workers are positioned inside the assembly line and can work in multiple stations. Therefore, the described line characteristics of the O-shaped line make it equal to a straight assembly line. Cars have optional features (options) such as several sensors, seat heating, sunroof, etc. These options are met by assembling additional parts to the harness, which is called different models of the harness. Sometimes, there are dozens of models of each harness which makes it impossible to build a separate assembly line for each model because of production space constraints and high assembly line initial building cost. To overcome this problem, the station is designed for the harness model, which has all options available, and all other models of the harness are assembled on the same assembly line. Although the production is multi-model, the manufacturing engineering department (ME) solves (re)balancing problem as it is a single model AL because they solve the problem for the harness model with the highest penetration (the harness model with the highest demand). The current line balance is broken because of several reasons, such as customer design change requests, material changes, ergonomic and production-related feedback from the production area, and changes in customer product demand. These changes may cause precedence diagram modifications, adding-removing tasks, changes in task processing times, and changes in cycle time. The proposed solution method and the software can rebalance the assembly line for all these changes, but in this study, its efficiency was tested by changing cycle times as it is the most common reason.

Table 1
Benchmark test problems and the results of the computational study

Test Problem					No. of Stations			CPU time (ms)		Smoothness Index		Line Efficiency		MSF		No. of Tasks Moved	
Precedence graph[1]	**No. of tasks**	**initial cycle time (s)**	**New cycle time (s)**	**m***	**h-GA[2]**	**Min.**	**Avg.**	**Avg.**	**Min.**	**Avg.**	**Min.**	**Avg.**	**Max.**	**Avg.**	**Max.**	**Avg.**	**Min.**
Arcus1	83	5408	5824	**14**	**-**	**14**	14	169	110	2584	1413	92.3	95	0.39	0.59	43.8	26
Arcus2	111	6016	5755	**27**	**-**	**27**	28.7	185	127	4524	1680	91.1	96.8	0.38	0.55	75.2	54
Barthold	148	513	564	**10**	11	**10**[a]	11	1303	817	511	3	90	99.9	0.27	0.46	104	70
Barthol2	148	101	95	**45**	-	46	47.2	792	643	75.6	36	97	97	0.14	0.23	136.3	120
Bowman	8	19	20	**5**	-	**5**	**5**	0.06	0	10.8	6.63	79.5	88	0.38	0.75	2.9	2
Buxey	29	30	27	**13**	-	**13**	14.2	6.8	3	20.1	9	84.2	92	0.26	0.72	21.9	8
Gunther	35	49	54	**9**	10	**9**[a]	10	13.4	8	24.6	14	89	99.4	0.28	0.63	20.6	11
Hahn	53	3507	2806	**6**	**6**	**6**	6	52.8	37	1917	1327	83.2	85	0.60	0.92	19.7	2
Heskiaoff	28	256	205	**5**	**5**	**5**	6	5.42	3	138	1	83.3	99	0.4	0.8	16.1	4
Jackson	11	9	7	**8**	-	**8**	8.5	0.17	0	5.1	4	73	82	0.71	0.82	6	4
Jaeschke	9	8	10	**4**	-	**4**	4.3	0.06	0	2.4	1	88.5	92.5	0.48	0.56	2.2	2
Kilbridge	45	111	92	**6**	-	**6**	7	36.6	24	74.3	39	85.3	88	0.35	0.74	25.5	8
Lutz1	32	1768	2020	**8**	**8**	**8**	11.9	8.51	4	1181	459	84.5	91	0.52	0.94	9.9	2
Lutz2	89	21	20	**25**	27	26[a]	26.9	196	132	14.7	8	90.3	93	0.3	0.51	65.7	45
Lutz3	89	79	83	**21**	-	**21**	22.7	169	116	76.4	31	87.2	94	0.45	0.8	66	52
Mansoor	11	94	62	**3**	-	**3**	3.9	0.2	0	30.2	1	79.2	99	0.51	0.91	4.8	2
Mertens	7	6	7	**5**	-	**5**	5.2	0.1	0	3.1	3	81.2	82	0.43	0.57	3.7	2
Mitchell	21	18	15	**8**	-	**8**	8.8	2.13	1	13.3	6	80	87.5	0.47	0.9	8.8	2
Mukherje	94	301	281	**16**	**16**	**16**	16	627	443	133	90	93.6	94.6	0.22	0.4	67.4	50
Roszieg	25	14	18	**8**	-	**8**	8	3.67	2	10.9	5	86	91	0.61	0.96	10.4	2
Sawyer	30	33	30	**12**	-	**12**	12.4	8.18	4	17.6	9	87.2	93	0.36	0.71	19.6	8
Scholl	297	1422	1515	**46**	51	48[a]	49.2	17150	13456	1495	1207	93.5	95.8	0.22	0.29	236	211
Tonge	70	185	176	**21**	-	**21**	22.7	115	86	151	55	91.6	93.5	0.34	0.68	40.2	16
Warnecke	58	86	92	**17**	20	18[a]	19.2	90	64	87.3	33	87.7	94	0.31	0.65	37.1	23
Wee-Mag	75	30	29	**63**	**-**	**63**	63.8	101	71	50.5	44	80.4	82	0.18	0.27	69.7	62

[1] benchmark data family of the precedence graph (Scholl, 1993), [2] results of the hybrid genetic algorithm of Belassiria et al.(2018), **m***: the minimal number of stations for the new cycle time, [a] the proposed method found a better solution than h-GA (Belassiria et al., 2018)

6.2. Analysis of the current assembly line

The IP harness assembly line was selected for our test case, which was balanced with a cycle time of 170 seconds. The selected model consists of 34 tasks, and the task precedence relationships are shared in Fig.2a. The diagram became very complicated when we showed all the relationships between tasks. To simplify the illustration, we combine the tasks into three groups and only show a relationship line between these groups. The first group contains tasks from 1 to 17. Each task in the first group is the predecessor of all tasks in the second group, which contains tasks from 18 to 29. Likewise, each task in the second group is the predecessor of all tasks in the third group, which contains tasks from 30 to 34. The task precedence relationship of task 1 is shared in Fig.2b as an example. Task 1 is the predecessor of tasks 18 to 29. Tasks 30 to 34 are the successors of the tasks from 18 to 29.

Nine workers are capable of working in the IP assembly line. The task processing times, shared in Table 2, differ from worker to worker. As some tasks need specific worker skills or training, they cannot be performed by some of the workers, which are stated with a dash in Table 2. The initial assignment of the assembly line is shared in Table 3.

Transporting tasks between stations may cause station modifications. This expense differs widely according to the line and task specifications. Task transportation cost is negligible for eight of the tasks, which are set to 0 EUR. For the remaining tasks, task transportation costs range from 134 EUR to 1889 EUR, which are shared in Table 2. The station operating cost was calculated as 2000 EUR for the current operation plan for eight weeks, including worker salary. Opening a new station needs ordering new equipment and hiring additional workers, which costs 3000 EUR, and closing a station costs 500 EUR.

The customer increased the demand for the harness, and cycle time is needed to be decreased from 170 seconds to 158 seconds to meet the new demand. There are seven objectives, which are described in section 3.3. The constraints of the problem are described in section 3.4. The software is set to stop after 500 iterations and run 10 times.

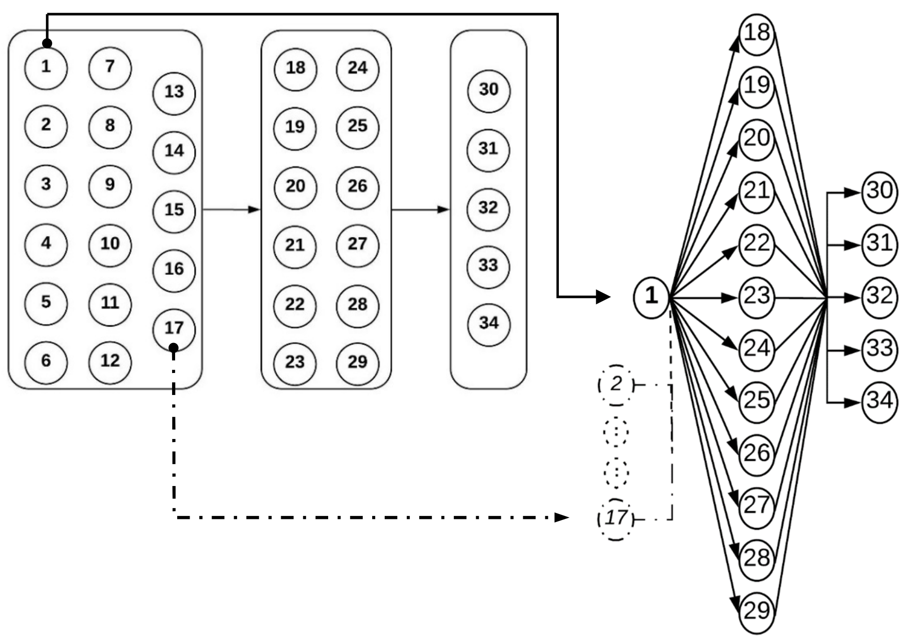

Fig.2a Precedence relations of the assembly line **Fig.2b** Precedence diagram of task 1

6.3. ErgoALWARBP decision support software: inputs, outputs, and solution environment

The ErgoALWARBP software was developed with C# by Visual Studio 2015. The software can read the inputs from an Excel workbook that is filled in the pre-specified format. The five different input categories and their content are as follows:

Initial data: Tasks (with names), workers, task precedence relations, the task processing time of each worker for each task (task-worker processing time matrix) (Table 2).

Work/time study: Task details (sub-tasks) including operation type and sub-task processing time information. Twelve operation types were defined and analyzed for specifying posture and force values of each operation that are used in OCRA ergonomic calculations (for OCRA input parameters, see Baykasoglu et al. (Baykasoglu et al., 2017): 114-115).

Kitting data: Technical move count variables and values of each task. These data are generated after a detailed analysis of technical drawings and assembly lines which are used in OCRA ergonomic calculations.
Initial assignment: Assembly line stations, station task assignment, station worker assignment, station time, cycle time.
New assembly line data: New cycle time, station operating/opening/closing costs, task transportation costs, operation duration (months), iteration, and run count (for calculations).

The software generates two outputs which are:

Task and worker assignment with ergonomic results of the case study assembly line (Table 3).
Solutions with objective value results (Table 4).

Table 2
Task transportation costs and task performing times of the workers in the industrial case study

Task number	Task transportation cost	Worker task times (seconds)								
		w1	w2	w3	w4	w5	w6	w7	w8	w9
1	668	-	20	**23**	22	25	20	24	26	21
2	239	15	**17**	15	-	16	16	19	20	16
3	443	14	**16**	14	19	15	17	13	18	-
4	1592	**28**	28	25	27	35	34	-	26	23
5	0	-	48	52	**50**	51	53	52	51	46
6	639	70	63	60	**64**	62	-	63	69	67
7	1537	8	**8**	-	11	7	10	5	9	10
8	0	**14**	-	16	16	12	18	-	13	11
9	0	31	38	30	**35**	38	-	31	-	39
10	598	53	-	**53**	52	52	56	52	56	51
11	720	**21**	25	19	20	-	20	22	20	21
12	524	**11**	-	14	12	10	10	8	14	9
13	392	**12**	12	10	14	15	10	-	11	-
14	1889	**52**	53	54	51	54	50	55	49	50
15	1307	42	**40**	38	37	40	40	-	38	47
16	680	84	92	**86**	-	85	82	88	84	87
17	0	75	**77**	-	77	76	79	74	73	82
18	0	16	19	18	**17**	15	16	18	16	18
19	1229	32	36	34	36	**34**	31	36	33	-
20	629	46	47	50	50	**49**	52	45	-	52
21	983	31	31	32	32	**33**	36	32	37	-
22	632	32	-	29	30	**30**	29	29	23	37
23	1052	11	7	-	9	**9**	7	10	9	10
24	134	21	25	18	25	-	**22**	21	21	22
25	821	52	53	58	62	51	**57**	-	65	57
26	850	11	13	13	13	-	**14**	16	16	14
27	0	-	6	6	5	-	**5**	4	4	5
28	475	65	58	55	50	53	**56**	-	60	56
29	473	7	11	10	8	12	**10**	12	-	10
30	0	34	34	31	-	34	41	**34**	30	34
31	971	19	17	15	15	17	-	**17**	17	21
32	1079	40	44	48	-	44	48	**44**	48	36
33	0	29	26	-	26	25	-	**26**	25	24
34	1431	46	49	48	49	-	47	**49**	52	52

6.4. Industrial case study implementation, results, and discussion

The software was run for 500 iterations on a Toshiba notebook with a 2.0 GHz AMD Turion Mobile TL-60 CPU. The station assignment details of the best solutions for each Goal are shared in Table 3. The solution names are given in the first column. In columns two and three, station numbers of the assembly line and assigned workers to each station are shared, respectively. In column four, the assigned tasks for each station are shared, which tasks are separated by commas. In columns five and six, each station's OCRA index and station time are shared, respectively. As an example of reading the results, there are seven stations in the initial assembly line, where worker 7(*w7*) was assigned to station 7(*s7*). The tasks 30, 31, 32, 33, 34 were assigned to *s7*, and *w7* can process these tasks in 170 seconds (station time). The *OCRA index* for *s7* is calculated as 10.99. In solution Goal-1, *w7* will still operate in *s7* and implement tasks 30, 32, 33, 34. So, while tasks 30, 32, 33, 34 will stay in *s7,* task 31 will be moved from *s7* to *s6* in the rebalanced AL.

Table 3
Initial assignment's and new solutions' station-task and station-worker assignments of the industrial case-study

Solutions	Station	Assigned Worker	Assigned Tasks	OCRA Index	Station Time
Initial Assignment	s1	w1	4, 8, 11, 12, 13, 14	9.66	138
	s2	w2	2, 3, 7, 15, 17	10.55	158
	s3	w3	1, 10, 16	9.99	162
	s4	w4	5, 6, 9, 18	14.63	166
	s5	w5	19, 20, 21, 22, 23	15.90	155
	s6	w6	24, 25, 26, 27, 28, 29	17.13	164
	s7	w7	30, 31, 32, 33, 34	10.99	170
Solution Goal-1 (min. Cost)	s1	w3	2, 9, 14, 13, 4, 11	11.20	153
	s2	w5	15, 12, 16, 3, 7	11.83	157
	s3	w4	10, 5, 1, 8	10.71	140
	s4	w2	6, 17, 29, 23	11.08	158
	s5	w1	21, 25, 20, 26, 18	24.65	156
	s6	w8	22, 19, 28, 27, 24, 31	11.90	158
	s7	w7	32, 33, 34, 30	10.74	153
Goal-2 (max. MSF)	s1	w6	10, 16, 1	10.25	158
	s2	w7	17, 5, 9	11.37	157
	s3	w1	12, 4, 13, 8, 14, 11, 2	9.49	153
	s4	w2	6, 3, 15, 7, 21	13.71	158
	s5	w4	22, 27, 29, 28, 18, 23, 26, 24	18.12	157
	s6	w5	25, 19, 20, 31	17.74	151
	s7	w8	33, 30, 34, 32	10.60	155
Goal-3 (max. worker MSF)	s1	w1	9, 4, 13, 11, 12, 3, 7, 8, 2	13.61	154
	s2	w8	17, 16	7.41	157
	s3	w3	6, 10, 1	10.68	136
	s4	w4	15, 5, 14, 18	12.44	155
	s5	w6	19, 22, 25, 27, 26, 24	20.57	158
	s6	w2	21, 28, 29, 20, 23	13.14	154
	s7	w7	34, 30, 32, 33	10.74	153
	s8	w9	31	10.77	21
Goal-4 (min. MAD of OCRA)	s1	w8	5, 16, 8, 7	13.04	157
	s2	w3	15, 6, 11, 9, 13	13.35	157
	s3	w5	4, 12, 1, 2, 10, 3	9.18	153
	s4	w1	14, 17, 26, 18	15.23	154
	s5	w4	24, 28, 19, 23, 21, 27	13.65	157
	s6	w9	22, 20, 29, 25	14.65	156
	s7	w7	32, 34, 30, 33	10.74	153
	s8	w2	31	13.31	17
Goal-5 (min. nr. of tasks moved)	s1	w3	13, 15, 9, 12, 3, 11, 1	10.78	148
	s2	w1	7, 17, 10, 2	11.98	151
	s3	w9	16, 4, 5	9.38	156
	s4	w5	8, 6, 14, 23, 18	12.30	152
	s5	w7	20, 19, 29, 27, 21, 22	15.52	158
	s6	w6	28, 25, 24, 26	17.58	149
	s7	w2	34, 32, 30, 31	12.65	144
	s8	w4	33	1.83	26
Goal-6 (max. line efficiency)	s1	w9	14, 10, 5, 12	10.84	156
Goal-7 (min. smoothness index)	s2	w6	16, 4, 13, 1, 7	10.99	156
	s3	w1	2, 11, 17, 9, 8	11.60	156
	s4	w3	6, 3, 15, 24, 27, 18	15.62	154
	s5	w2	21, 29, 28, 20, 23	13.14	154
	s6	w4	25, 22, 19, 26, 31	18.62	156
	s7	w7	30, 33, 34, 32	10.74	153

The same solution was found for min. LE (G6) and min. SX (G7). The SX values range between 4.12 and 25.44 (except for one solution with SX=33.08) for LE values higher than 95.75. These high-performance results are very close to each other, which is better than the initial solution. For lower efficiencies, the SX dropped around 100 and ranged widely between 75 and 145, which became not dependent on LE values. The relation between SX and LE values of the 500 feasible solutions is shared in Fig.3.

Fig. 3. Line efficiency & line smoothness index values of the feasible solutions

Computational results of the best solutions for the seven goals, the average results, and the standard deviation values of all feasible solutions are shared in Table 4.

In the best solution for Goal-1 ($\boldsymbol{min.}\, TC_{rb}$), 15 tasks were transported, which had a 7471 EUR rebalancing cost. This cost is 54% less than solution Goal-2 (max. MSF), 57% less than solution Goal-4 ($min.\, madOCRA$), 42% less than solution Goal-5 (min. task movement), and 55% less than the average cost of all feasible solutions. Although it has better results in terms of the LE and SX in comparison with the initial assignment, there is a 13% increase in the OCRA result, which is the worst station ergonomics between the solutions. The distribution of solutions in terms of the total cost of rebalancing is shared in Fig.4.

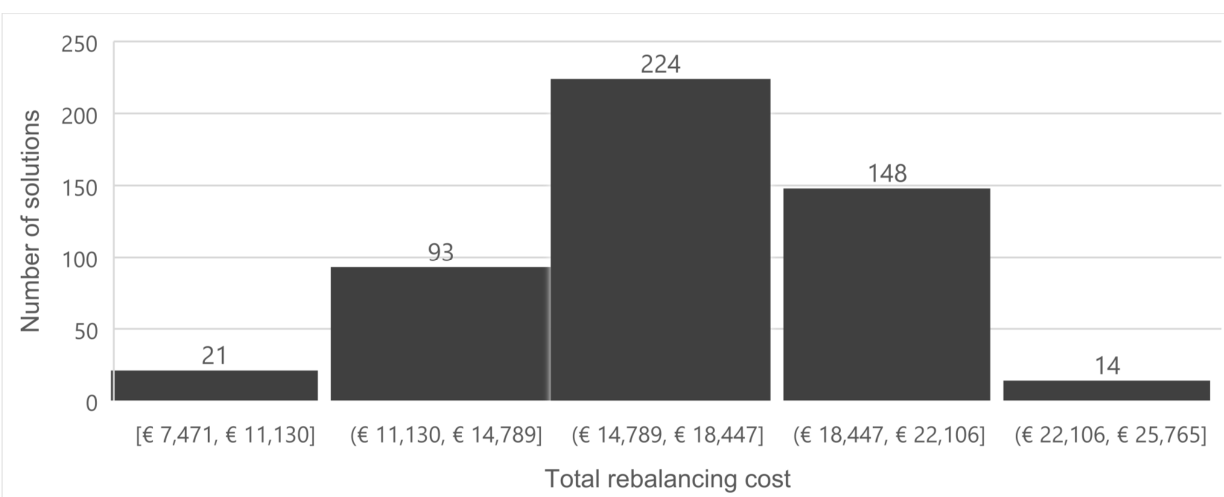

Fig.4 The distribution of the feasible solutions in terms of total rebalancing cost

Table 4
Computational results of the industrial case study

Solution	CPU time (ms)	No. of stations	Cycle time (s)	Total rebalancing cost (EUR)	Task movement cost (EUR)	No. of tasks moved	MSF	Worker MSF	OCRA MAD	Line efficiency	Smoothness index
Initial assignment	-	7	170	-	-	-	-	-	2.74	93.53	38.85
Goal1- min. rebalancing cost	209	7	158	**7471**	7471	15	0.31	0.18	3.10	97.20	19.47
G2- max. MSF	288	7	158	16333	16333	25	**0.56**	0.32	2.99	98.46	9.22
G3- max. worker MSF	207	8	158	17174	12383	16	0.44	**0.43**	2.52	86.08	139.00
G4- min. MAD of OCRA indexes	239	8	157	17346	12346	20	0.28	0.12	**1.47**	87.90	140.15
G5- min. no. of tasks moved	211	8	158	13022	8022	**13**	0.38	0.15	3.13	85.76	133.75
G6- max. line efficiency G7- min. smoothness index	196	7	156	10553	10533	20	0.30	0.16	2.33	**99.36**	**4.12**
Average of 500 feasible solutions	211.2	7.57	157.6	16879	6650	20.96	0.31	0.15	3.08	89.92	89.84
Standard deviation	15.47	0.46	0.72	3044	2500	2.87	0.05	0.08	0.54	5.01	50.59

The solution with the best ergonomics (G4) has 46% better ergonomics than the initial solution. However, it had the worst rebalancing cost with 17346 EUR and got worse LE and SX compared to the initial assignment. Also, the rebalancing cost will become higher because of the operational cost ($cost_{run}$) of its additional station, if the planning period will be longer. This led the company not to choose this solution.

Although it costs 3K EUR more than the min. cost solution, the company chose to implement the best solution for G6 and G7. The solution results are analyzed as below:

- With a 6% improvement, the *LE* is close to the optimum efficiency value of 100% with LE=99.36%.
- The *SX* is decreased by 89% to 4.12, close to optimum SX value 1.
- Although having 36% worse OCRA result than the solution with the best ergonomics (G4), the solution improves line ergonomics by 14.9% compared to the initial assignment.
- The solution has very close results to the minimum cost solution (G1) in terms of MSF (3% lower) and worker MSF (11% lower).

After implementing the suggested method in the real-world assembly line and analyzing the solutions, we can conclude that it is crucial to decide the order of importance of the objectives (the goals) while choosing the best solution. The planning period and the number of stations of the rebalanced line are essential factors while calculating the total cost of rebalancing. The operating cost ($cost_{run}$) changes in direct relation with the planning period. As it is shared in Eq. 1, the change in the no. of stations is the multiplier of $cost_{run}$. The total cost can change a lot in more extended planning periods if the number of stations is changed in the rebalanced AL. Also, the number of tasks moves has a low effect on total rebalancing cost if tasks with the low moving cost are selected for movement. Therefore, it is essential to define the planning period well if the cost has a high priority for the company.

With the ErgoALWARBP solution method and software, several benefits are gained in the company for solving the AL rebalancing problem. The most valuable gains can be summarized as below:

- Method engineers were manually solving the line rebalancing problem before the ErgoALWARBP software. Therefore, the problem-solving speed and the solution quality varied by the experience level of the engineers. The dependency on the experience level of engineers and the risk of making calculation errors were eliminated with the help of our solution method and the decision support software.
- We considered seven objectives in our solution method, described in section 3.3. Method engineers can manually solve the rebalancing problem in 4-6 weeks/engineer according to the scope of the change request and their experience level, although they can only consider a smaller number of objectives and constraints simultaneously. In a frequently changing agile work environment, the customer may request several product changes during the solution implementation, which causes a lot of time and material waste. The ErgoALWARBP software can calculate and propose several effective solutions that meet different goals in just seconds. Thus, the software improved the team's agility for the customer change requests by increasing the solution speed at least ten thousand times, decreasing the rebalancing cost, and eliminating the cost of wastes.
- Manually calculating line ergonomics needs so much time and effort, which caused the team to ignore ergonomic criteria during the initial rebalancing. It was usually preferred to solve ergonomic problems according to feedback from the production area after solution implementation. With the help of ErgoALWARBP software, the team could find solutions with better ergonomics, considering all ergonomic criteria. This helped to decrease the costs that were related to additional ergonomic improvements and quickly improve the assembly line's ergonomic conditions, which may cause musculoskeletal disorders.

7. Conclusion

In this study, a novel rebalancing problem about workers' specific skills is studied under ergonomic aspects, the ErgoALWARBP. A formal definition of the problem is stated, which is multi-objective by nature. The objectives are maximizing the similarity of the assigned tasks to the stations, the similarity of performed tasks by the workers, and the line efficiency while minimizing the total cost of rebalancing, line smoothness index, ergonomic objective, and the number of relocated tasks. To deal with the novel ErgoALWARBP, a randomized constructive rule-based heuristic approach is developed, and a preemptive goal programming approach is applied. To evaluate the performance of the developed rebalancing method, it was tested on the well-known benchmark data. Experimental results showed the efficiency of the developed algorithm. Then, the solution method is applied by a company that manufactures automotive harness by using the developed decision support software. The software generated several solutions, and they were evaluated by the method engineers. The most fitting solution was chosen and implemented on the assembly line among the suggested alternatives. Specifically, the proposed systematic approach improves problem-solving speed, assembly line ergonomics, line efficiency, and line smoothness index, with less rebalancing cost. The developed ErgoALWARBP software improved the organization's agility for frequent changes, which may lead to the need for rebalancing, and decreased the dependency of the solution quality to the engineer's experience level. Future researches can extend the ErgoALWARBP and the heuristic solution method to solve U-type/two-sided line configurations, multi/mixed-model assembly lines, and stochastic worker availability/task times.

Code and Data Availability

The source codes of the C# windows application and data can be shared upon request.

References

Ağpak, K. (2010). An approach to find task sequence for re-balancing of assembly lines. *Assembly Automation*, *30*(4), 378–387. https://doi.org/10.1108/01445151011075834

Akyol, S. D., & Baykasoğlu, A. (2019a). ErgoALWABP: a multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem under ergonomic risk factors. *Journal of Intelligent Manufacturing*, *30*(1), 557–573. https://doi.org/10.1007/s10845-016-1246-6

Akyol, S. D., & Baykasoğlu, A. (2019b). A multiple-rule based constructive randomized search algorithm for solving assembly line worker assignment and balancing problem. *Journal of Intelligent Manufacturing*, *30*(2), 557–573. https://doi.org/10.1007/s10845-016-1262-6

Altemeier, S., Helmdach, M., Koberstein, A., & Dangelmaier, W. (2010). Reconfiguration of assembly lines under the influence of high product variety in the automotive industry - A decision support system. *International Journal of Production Research*, *48*(21), 6235–6256. https://doi.org/10.1080/00207540903252290

Arcus, A. L. (1966). COMSOAL A Computer Method of Sequencing Operations for Assembly Lines. *International Journal of Production Research*, *4*(4), 259–277. https://doi.org/10.1080/00207546508919982

Baykasoglu, A., Tasan, S. O., Tasan, A. S., & Akyol, S. D. (2017). Modeling and solving assembly line design problems by considering human factors with a real-life application. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *27*(2), 96–115. https://doi.org/10.1002/hfm.20695

Belassiria, I., Mazouzi, M., ELfezazi, S., Cherrafi, A., & ELMaskaoui, Z. (2018). An integrated model for assembly line re-balancing problem. *International Journal of Production Research*, *56*(16), 5324–5344. https://doi.org/10.1080/00207543.2018.1467061

Borba, L., & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the Assembly Line Worker Assignment and Balancing Problem. *Computers & Operations Research*, *45*, 87–96. https://doi.org/10.1016/j.cor.2013.12.002

Celik, E., Kara, Y., & Atasagun, Y. (2014). A new approach for rebalancing of U-lines with stochastic task times using ant colony optimisation algorithm. *International Journal of Production Research*, *52*(24), 7262–7275. https://doi.org/10.1080/00207543.2014.917768

Chaves, A. A., Lorena, L. A. N., & Miralles, C. (2007). Clustering search approach for the assembly line worker assignment and balancing problem. *Proceedings of the 37th International Conference on Computers and Industrial Engineering*, 1469–1478.

Corominas, A., Pastor, R., & Plans, J. (2008). Balancing assembly line with skilled and unskilled workers. *Omega*, *36*(6), 1126–1132. https://doi.org/10.1016/j.omega.2006.03.003

Costa, A. M., & Miralles, C. (2009). Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics*, *120*(2), 625–632. https://doi.org/10.1016/j.ijpe.2009.04.013

Fattahi, P., & Samouei, P. (2016). An Analytical Approach for Single and Mixed-Model Assembly Line Rebalancing and Worker Assignment Problem. *Journal of Industrial and Systems Engineering*, *9*(1), 79–91. http://www.jise.ir/article_11811_9.html%5Cnhttp://jise.ir/article_11811_2014.html

Fichera, S., Costa, A., & Cappadonna, F. A. (2017). Heterogeneous workers with learning ability assignment in a cellular manufacturing system. *International Journal of Industrial Engineering Computations*, *8*(4), 427–440. https://doi.org/10.5267/j.ijiec.2017.3.005

Gamberini, R., Gebennini, E., Grassi, A., & Regattieri, A. (2009). A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem. *International Journal of Production Research*, *47*(8), 2141–2164. https://doi.org/10.1080/00207540802176046

Gamberini, R., Grassi, A., & Rimini, B. (2006). A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *International Journal of Production Economics*, *102*(2), 226–243. https://doi.org/10.1016/j.ijpe.2005.02.013

Girit, U., & Azizoğlu, M. (2021). Rebalancing the assembly lines with total squared workload and total replacement distance objectives. *International Journal of Production Research*, *59*(22), 6702–6720. https://doi.org/10.1080/00207543.2020.1823027

Grangeon, N., Leclaire, P., & Norre, S. (2011). Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research*, *49*(22), 6609–6628. https://doi.org/10.1080/00207543.2010.539025

Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., & Chan, S. F. (2008). A Genetic-Algorithm-Based Optimization Model for Solving the Flexible Assembly Line Balancing Problem With Work Sharing and Workstation Revisiting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(2), 218–228. https://doi.org/10.1109/TSMCC.2007.913912

Hignett, S., & McAtamney, L. (2000). Rapid Entire Body Assessment (REBA). *Applied Ergonomics*, *31*(2), 201–205. https://doi.org/10.1016/S0003-6870(99)00039-3

Ishigaki, A. (2018). Heuristic approach and application for rebalancing of a mixed-model assembly line. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *12*(3), 1–10. https://doi.org/10.1299/jamdsm.2018jamdsm0078

Lai, R., Hou, L., & Wu, Y. (2015). Evolution Balancing of the Small-Sized Wheel Loader Assembly Line. *Shock and Vibration*, *2015*, 1–11. https://doi.org/10.1155/2015/213159

Li, G., & Buckle, P. (1999). Current techniques for assessing physical exposure to work-related musculoskeletal risks, with emphasis on posture-based methods. *Ergonomics*, *42*(5), 674–695. https://doi.org/10.1080/001401399185388

Li, Y. (2017). The type-II assembly line rebalancing problem considering stochastic task learning. *International Journal of Production Research*, *55*(24), 7334–7355. https://doi.org/10.1080/00207543.2017.1346316

Li, Y., & Boucher, T. O. (2017). Assembly line balancing problem with task learning and dynamic task reassignment. *The International Journal of Advanced Manufacturing Technology*, *88*(9–12), 3089–3097. https://doi.org/10.1007/s00170-016-9014-5

Li, Y., Li, Z., & Saldanha-da-Gama, F. (2021). New approaches for rebalancing an assembly line with disruptions. *International Journal of Computer Integrated Manufacturing*, 1–18. https://doi.org/10.1080/0951192X.2021.1925967

Liu, C., Li, W., Lian, J., & Yin, Y. (2012). Reconfiguration of assembly systems: From conveyor assembly line to serus. *Journal of Manufacturing Systems*, *31*(3), 312–325. https://doi.org/10.1016/j.jmsy.2012.02.003

Makssoud, F., Battaïa, O., Dolgui, A., Mpofu, K., & Olabanji, O. (2015). Re-balancing problem for assembly lines: new mathematical model and exact solution method. *Assembly Automation*, *35*(1), 16–21. https://doi.org/10.1108/AA-07-2014-061

Makssoud, F., Battäıa, O., Dolgui, A., Battaïa, O., & Dolgui, A. (2013). An Exact Method for the Assembly Line Re-balancing Problem. *International Conference on Advances in Production Management Systems*, *414*, 159–166. https://doi.org/10.1007/978-3-642-41266-0_20

Mardani-Fard, H. A., Hadi-Vencheh, A., Mahmoodirad, A., & Niroomand, S. (2020). An effective hybrid goal programming approach for multi-objective straight assembly line balancing problem with stochastic parameters. *Operational Research*, *20*(4), 1939–1976. https://doi.org/10.1007/s12351-018-0428-8

McAtamney, L., & Nigel Corlett, E. (1993). RULA: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, *24*(2), 91–99. https://doi.org/10.1016/0003-6870(93)90080-S

Miralles, C., García-Sabater, J. P., Andrés, C., & Cardós, M. (2008). Branch and bound procedures for solving the Assembly Line Worker Assignment and Balancing Problem: Application to Sheltered Work centres for Disabled. *Discrete Applied Mathematics*, *156*(3), 352–367. https://doi.org/10.1016/j.dam.2005.12.012

Mokhtari, H., & Mozdgir, A. (2015). A Bi-Objective Approach for Design of an Assembly Line Re-Balancing System: Mathematical Model and Differential Evolution Algorithms. *International Journal of Industrial Engineering & Production Research*, *26*(2), 105–117. http://ijiepr.iust.ac.ir/article-1-613-en.html

Moreira, M. C. O., Miralles, C., & Costa, A. M. (2015). Model and heuristics for the Assembly Line Worker Integration and Balancing Problem. *Computers & Operations Research*, *54*, 64–73. https://doi.org/10.1016/j.cor.2014.08.021

Occhipinti, E. (1998). OCRA: A concise index for the assessment of exposure to repetitive movements of upper limbs. *Ergonomics*, *41*, 1290–1311.

Oliveira, F. S., Vittori, K., Russel, R. M. O., & Travassos, X. L. (2012). Mixed assembly line rebalancing: A binary integer approach applied to real world problems in the automotive industry. *International Journal of Automotive Technology*, *13*(6), 933–940. https://doi.org/10.1007/s12239-012-0094-4

Otto, A., & Scholl, A. (2011). Incorporating ergonomic risks into assembly line balancing. *European Journal of Operational Research*, *212*(2), 277–286. https://doi.org/10.1016/j.ejor.2011.01.056

Özbakır, L., & Seçme, G. (2020). A hyper-heuristic approach for stochastic parallel assembly line balancing problems with equipment costs. *Operational Research*. https://doi.org/10.1007/s12351-020-00561-x

Rahman, C. L. (2010). Re-balancing of Generalized Assembly Lines – Searching Optimal Solutions for SALBP. *International Conference on Industrial Engineering and Operations Management*, *January*, 3–8. https://doi.org/10.13140/2.1.1620.7047

Ramezanian, R., & Khalesi, S. (2021). Integration of multi-product supply chain network design and assembly line balancing. *Operational Research*, *21*(1), 453–483. https://doi.org/10.1007/s12351-019-00453-9

Sancı, E., & Azizoğlu, M. (2017). Rebalancing the assembly lines: exact solution approaches. *International Journal of Production Research*, *55*(20), 5991–6010. https://doi.org/10.1080/00207543.2017.1319583

Schaub, K., Caragnano, G., Britzke, B., & Bruder, R. (2013). The European Assembly Worksheet. *Theoretical Issues in Ergonomics Science*, *14*(6), 616–639. https://doi.org/10.1080/1463922X.2012.678283

Scholl, A. (1993). Data of assembly line balancing problems. *Schriften Zur Quantitativen Betriebswirtschaftslehre*, *16/93*.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines* (2nd ed., p. 318). Physica. https://ideas.repec.org/p/dar/wpaper/10881.html

Serin, F., Mete, S., & Çelik, E. (2019). An efficient algorithm for U-type assembly line re-balancing problem with stochastic task times. *Assembly Automation*, AA-07-2018-106. https://doi.org/10.1108/AA-07-2018-106

Sikora, C. G. S., Lopes, T. C., & Magatão, L. (2017). Traveling worker assembly line (re)balancing problem: Model, reduction techniques, and real case studies. *European Journal of Operational Research*, *259*(3), 949–971. https://doi.org/10.1016/j.ejor.2016.11.027

Tóth, N., & Kulcsár, G. (2021). New models and algorithms to solve integrated problems of production planning and control taking into account worker skills in flexible manufacturing systems. *International Journal of Industrial Engineering Computations*, *12*(4), 381–400. https://doi.org/10.5267/j.ijiec.2021.5.004

Waters, T. R. (1994). *Applications Manual for the Revised NIOSH Lifting Equation*. U.S. Department of Health and Human Services, Public Health Service, Centers for Disease Control and Prevention, National Institute for Occupational Safety and Health, Division of Biomedical and Behavioral Science.

Yang, C., & Gao, J. (2016). Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research*, *65*, 139–148. https://doi.org/10.1016/j.cor.2015.07.007

Yang, C., Gao, J., & Sun, L. (2013). A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, *65*(1), 109–116. https://doi.org/10.1016/j.cie.2011.11.033

Zacharia, P. T., & Nearchou, A. C. (2016). A population-based algorithm for the bi-objective assembly line worker assignment and balancing problem. *Engineering Applications of Artificial Intelligence*, *49*, 1–9. https://doi.org/10.1016/j.engappai.2015.11.007

Zha, J., & Yu, J. J. (2014). A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment. *Journal of Manufacturing Systems*, *33*(1), 93–102. https://doi.org/10.1016/j.jmsy.2013.08.002

Zhang, Y.-H. Y., Hu, X.-F. X., & Wu, C.-X. C. (2018). Push Rule-Based Heuristic Method for Multi-objective Two-Sided Assembly Line Rebalancing Problem. *DEStech Transactions on Engineering and Technology Research*, *pmsms*. https://doi.org/10.12783/dtetr/pmsms2018/24879

Zhang, Y., Hu, X., & Wu, C. (2018). A modified multi-objective genetic algorithm for two-sided assembly line re-balancing problem of a shovel loader. *International Journal of Production Research*, *56*(9), 3043–3063. https://doi.org/10.1080/00207543.2017.1402136

Zhang, Y., Hu, X., & Wu, C. (2020). Improved imperialist competitive algorithms for rebalancing multi-objective two-sided assembly lines with space and resource constraints. *International Journal of Production Research*, *58*(12), 3589–3617. https://doi.org/10.1080/00207543.2019.1633023

Appendix

OCRA index is calculated as in the following:

$$OCRA = Actual\ frequency\ /\ Recommended\ frequency \tag{A.1}$$

$$Actual\ frequency\ = \frac{Number\ of\ technical\ actions}{station\ time} * 60 \tag{A.2}$$

$$Recommended\ frequency\ \ = CF * PM * FM * RM * ARF * (RcM * DuM) \tag{A.3}$$

$$avgOCRA = \frac{sum\ of\ OCRA\ indexes\ of\ all\ stations}{number\ of\ stations} \tag{A.4}$$

$$madOCRA = \sqrt{\frac{(avgOCRA - OCRA_a)^2}{m-1}} \tag{A.5}$$

OCRA index is calculated for each station after the assignment of tasks and worker in that particular station has completed. Details of the OCRA index parameters can be examined from Akyol and Baykasoğlu (2019a) and Otto and Scholl (2011).