Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi
Pamukkale University Journal of Engineering Sciences

A hybrid genetic algorithm for solving energy-efficient mixed-model robotic two-sided assembly line balancing problems with sequence-dependent setup times

Enerji verimli sıra-bağımlı hazırlık zamanlı karışık modelli robotik çift taraflı montaj hattı dengeleme problemlerinin çözümü için bir hibrit genetik algoritma

Şehmus ASLAN[1*]

[1]Mardin Artuklu University, Mardin, Turkey.
sehmusaslan@artuklu.edu.tr

Received/Geliş Tarihi: 04.10.2023 Revision/Düzeltme Tarihi: 05.01.2024 doi: 10.5505/pajes.2024.17092
Accepted/Kabul Tarihi: 08.02.2024 Research Article/Araştırma Makalesi

Abstract

Serious environmental challenges such as global warming and climate change have captured a growing amount of public awareness in the last decade. Besides monetary incentives, the drive for environmental preservation and the pursuit of a sustainable energy source have contributed to an increased recognition of energy usage within the industrial sector. Meanwhile, the challenge of energy efficiency stands out as a major focal point for researchers and manufacturers alike. Efficient assembly line balancing plays a vital role in enhancing production effectiveness. The robotic two-sided assembly line balancing problem (RTALBP) commonly arises in manufacturing facilities that produce large-sized products in high volumes. In this scenario, multiple robots are placed at each assembly line station to manufacture the product. The utilization of robots is extensive within two-sided assembly lines, primarily driven by elevated labour expenses. However, this adoption has resulted in the challenge of increasing energy consumption. Therefore, in this study, a new hybrid genetic algorithm is introduced, incorporating an adaptive local search mechanism. for the mixed-model robotic two-sided assembly line balancing problems with sequence-dependent setup times. This algorithm has two main objectives: minimizing cycle time (time-based approach) and overall energy consumption (energy-based approach). Depending on managerial priorities, either the time-based or energy-based model can be chosen for different production timeframes.

Keywords: Robotic Two-Sided, Assembly Line, Energy Consumption, Hybrid Genetic Algorithm, Setup Times

Öz

Son on yılda küresel ısınma ve iklim değişikliği gibi çevresel sorunlar, kamuoyunun giderek dikkatini çekmiştir. Parasal teşviklerin yanısıra, çevresel koruma ve sürdürülebilir enerji kaynağı arayışı nedeniyle endüstride enerji kullanımı daha da önem arz etmiştir. Aynı zamanda, enerji verimliliği sorunu, araştırmacılar ve üreticiler için de önemli bir odak noktası olarak öne çıkmaktadır. Verimli montaj hattı dengeleme, üretim etkinliğini artırmada önemli bir rol oynamaktadır. Robotik çift taraflı montaj hattı dengeleme problemi (RÇMHDP), yüksek hacimde büyük ürünler üreten üretim tesislerinde yaygın olarak karşılaşılan bir problemdir. Bu montaj hattında, ürünü üretmek için her montaj hattı istasyonunda birden fazla robot bulunur. İki taraflı montaj hatlarında robotların kullanımı, özellikle yüksek işçilik maliyetleri nedeniyle yaygın bir şekilde tercih edilmektedir. Ancak, bu durumda da enerji maliyetleri sorunu ortaya çıkmaktadır. Bu nedenle bu çalışmada; sıra-bağımlı hazırlık zamanlı karışık modelli robotik çift taraflı montaj hattı dengeleme problemleri için, uyarlanabilir yerel arama mekanizmasını içeren yeni bir hibrit genetik algoritma önerilmiştir. Bu algoritmanın iki ana amacı vardır: çevrim süresini (zamana dayalı yaklaşım) ve toplam enerji tüketimini (enerjiye dayalı yaklaşım) en aza indirmek. Yönetimsel önceliklere bağlı olarak, farklı üretim zaman dilimleri için zamana dayalı veya enerjiye dayalı model seçilebilir.

Anahtar Kelimeler: Robotik Çift Taraflı, Montaj Hattı, Enerji Tüketimi, Hibrit Genetik Algoritma, Hazırlık Zamanları

1 Introduction

Nowadays, one of the most important topic is manufacturing with efficient energy due to climate change. In order to reduce the impact of climate change, United Nations members states signed with defined target in the Paris Climate Agreement. The one of the main targets is that limitation of average global warming with 2 °C above preindustrial temperature [1]. A significant portion of greenhouse gas emissions, which is one of the factors causing global warming, comes directly from the manufacturing industry. That's why carbon emissions are restrictively limited for manufacturing industries in many countries. On the other hand, energy and energy costs for manufacturers are also an important key to global competition. The manufacturing industry has a strong focus on energy reduction and the creation of an environmentally friendly environment as a result of the increase in energy costs. [2]. Therefore, when both energy saving and economic criteria are considered at the same time, the development of energy efficient manufacturing technologies is of paramount importance. Hence, the improvement of assembly profits and the reduction of energy consumption have become important and challenging research topics that require attention. Energy consumption and costs are high in the manufacturing industry, particularly in places like assembly lines. Companies with an assembly line used effectively and efficient energy can gain many economic advantages and it facilitates competition the companies with others.

Assembly lines are a widely recognized research area in the production environment that has been studied under

*Corresponding author/Yazışılan Yazar

numerous titles for almost 70 years. Essentially, assembly lines are a collection of workstations that are arranged sequentially where specific tasks, typically involving assembly operations, are performed. Assembly lines face the primary challenge of the assembly line balancing problem (ALBP), which involves assigning and organizing tasks in a specific order across multiple workstations, while adhering to precedence relation constraints and optimizing objectives such as minimizing workstations, cycle time, and costs. This is a well-known production problem, and numerous studies have been conducted on this subject [3]. The survey papers of Boysen et al. [4] and Sivasankaran and Shahabudeen [5] can be analyzed.

Based on the number of products in a line, ALBP is classified into three different parts which are single-model (SALBP), mixed-model (MMALBP) and multi-model (MuMALBP). SALBP is core of the ALBP, and it can be defined as a single product is manufactured in a line. In the MMALBP, however, different models of product which are in the same product family are produced on the same line simultaneously. In other words, it is a complex version of the SALBP. On the other hand, MuMALBP problem can be described that varying model which are not generally allow to negligible set-up time due to similarity of process are manufactured with batches in same line. The key issue of MuMALBP is lot sizing because this problem is linked to balancing and batch sequencing in the multi-model line [6]. As far as task time is concerned, ALBP is categorized usually four parts which are deterministic, dynamic, stochastic and fuzzy task time in the literature. Deterministic task time is that task time is accepted static and known under all conditions. Dynamic task time is defined that there is a systematic variation in the task time due to certain factors, such as learning effects [7]. While task time variation is defined by a probability distribution, the problem is called as ALBP with stochastic task time. In certain cases, this variation can be represented by fuzzy task time. Furthermore, line design layout is another important classification in the ALBP literature. Two-sided assembly line (TSAL), straight line, U-type line and parallel line are well-known line design.

TSALs, or Two-Sided Assembly Lines, have gained popularity in recent years due to their ability to produce high volumes and large-sized products. They also allow workers to work on opposite sides of the product simultaneously, which is especially useful for the assembly of trucks, buses, and automobiles. The analysis of this line configuration was first presented in the literature by Bartholdi [8]. TSALs can be defined that it is an arranged of sequential workstations in which tasks can be completed at the two side of the line, left and right. A configuration instance of TSAL layout is shown in Fig. 1. It ensures opportunity performing of two workers on same product simultaneously. By considering this reason, balancing problem is differentiated from one-sided ALs, known as TSAL balancing problem (TALBP).

Figure 1. Two-sided assembly line.

TALBP is a new subject of concern, and the numbers of studies are limited when it is compared with ALBP. Li et al. [9] can be investigated for more detail on TSALBP and solution methods.

The rapid advancement of technology has resulted in the widespread usage of robots in various fields, including assembly line systems. In assembly lines, robots provide several manufacturing flexibility and safety, less reliance on specialized personnel, and lower energy consumption [10], [11]. There are many studies on the Robotic Assembly Line Balancing Problem (RALBP) in the literature. Rubinovitz et al. [12] were the first to introduce RALBP in the literature, proposing a branch and bound algorithm for solving and designing the problem. Then, Aghajani et al. [13] proposed a mixed-integer programming (MIP) model and simulated annealing algorithm (SA) for minimizing mixed-model RALBP's cycle time. In their study, Çil et al. [14] then set out a mathematical model and beam search algorithm as a means to decrease the overall cycle time for RALBP. A recent paper by Chutima [15] presents an extensive literature review on RALBP. However, few studies address energy consumption in the literature. Nilakantan et al. [16] published the first paper in the literature that examined energy consumption on RALBP. Simultaneous proposals were made for nonlinear mathematical models aimed at reducing cycle time and energy usage. Li et al. [17] developed an MIP model aimed at reducing both cycle time and energy consumption in two-sided RALBP. A restarted simulated annealing algorithm is proposed, specifically designed to address the challenges posed by large-scale problems. In their study, Nilakantan et al. [18] proposed a model aiming to enhancing the operational efficiency of robotic assembly line systems, while concurrently reducing their carbon footprint. A multi-objective co-operative coevolutionary (MOCC) method was designed to address this problem. Zhang et al. [19] proposed an innovative non-linear model for U-type RALBP with the aim of mitigating noise emission, reducing cycle time as well as minimizing emissions of carbon. Zhang et al. [20] also developed a mathematical model that incorporates several objectives and utilized a Pareto-based Artificial Bee Colony algorithm (PBABC) to optimize the U-type RALBP process, with the aim of simultaneously reducing cycle time and energy consumption. In their study, Zhou and Wu, [11] consider productivity-related targets (total working load) and green manufacturing objectives (total energy usage) for RALBP. Baş et al. [21] suggested a novel integer programming model in order to minimize cycle time of a Dishwasher Factory consisting mixed-model U-type robotic assembly line. In RALBP literature, the studies that consider energy consumption are summarized and given in Table 1.

In addition to the literature given above, Aslan [22] proposed a mathematical model and a variable neighborhood search algorithm for type II mixed-model robotic two-sided assembly line balanbing problems with sequence dependent setup times (MRTALBPS-II). However, it is determined that there is not any published paper that minimize both cycle time and total energy consumption for MRTALBPS in the literature. In order to close this gap, this paper aims to develop two models with dual focus for MRTALBPS, which are time-based model and energy-based model. The primary objective of time based model is to minimize cycle time and secondary objective is to minimize total energy consumption of MRTALBPS. On the other hand, the primary objective of energy based model is to minimize total energy consumption and secondary objective to minimize cycle time of MRTALBPS. The main contribution of this paper can be given as follows:

1. In this paper, a new objective function is added to mixed integer programming (MIP) model of Aslan [22] in order to calculate total energy consumption of MRTALBPS-II,
2. An effective hybrid genetic algorithm (hGA) with adaptive local search scheme is proposed in order to solve large instances in reasonable CPU time. Adaptive local search scheme prevents premature convergence by inserting new individuals with certain high fitness values into current hGA loop.
3. This paper represents a significant contribution to the existing body of literature, as most prior studies in the field of robotic assembly line balancing have primarily concentrated on minimizing cycle time and reducing costs, with little attention paid to optimizing energy consumption. The primary objective of this research is to create a hGA for simultaneously minimizing cycle time and the total energy usage within a robotic assembly line.
4. A comparative analysis is conducted to evaluate the performance of the hGA and VNS algorithm proposed in Aslan [22]. Statistical analysis reveals that the hGA proposed in this study outperforms VNS algorithm in terms of overall performance.

The rest of the paper organized as follows: In section 2, mixed-model robotic two-sided assembly line balancing problems with sequence-dependent setup times are defined. Proposed hybrid genetic algorithm is explained in section 3. Computational results are analyzed detail in section 4. Finally, section 5 concludes the study and provide suggestions for future research.

Table 1. Summary of RALBP literature considering energy consumption

Papers	**Number of Models**	**Line Type**	**Objectives**	**Solution Technique**
Nilakantan et al. [16]	Single	Straight	Min. cycle time and total energy consumption	Particle Swarm Optimization
Li et al. [17]	Single	Two-Sided	Min. cycle time and energy consumption	Restarted Simulated Annealing
Nilakantan et al. [18]	Single	Straight	Min. the carbon footprint, and max. of line efficiency	Multi-objective co-operative co evolutionary algorithm
Zhang et al. [19]	Single	U-shaped	Min. carbon emission, noise emission and cycle time	Hybrid Pareto Grey Wolf Optimization
Zhang et al. [20]	Single	U-shaped	Min. of cycle time and energy consumption	Pareto artificial bee colony algorithm
Zhou and Wu [11]	Single	Straight	Min. the number of workstations and total energy consumption	Decomposition-based multi-objective algorithm
Sun et al. [2]	Single	Straight	Min. cycle time and total energy consumption	Bound-guided hybrid estimation of distribution algorithm
Soysal-Kurt and İşleyen [10]	Single	Parallel	Min. cycle time and total energy consumption	A Pareto hybrid discrete firefly algorithm
Proposed Study	**Mixed**	**Two-Sided**	**Min. cycle time and total energy consumption**	**Hybrid Genetic Algorithm**

2 Problem definition and formulation

1.1 Problem definition

In order to minimize cycle time while meeting mated station number, direction, and precedence constraints, the MRTALBPS-II assigns one of the R available robots to each station and distributes a set of I jobs to a set of J mated-stations. In MRTALBPS-II, robots at a specific mated station must assemble similar product models simultaneously. Although each model's tasks have a set of precedence relationships, the models' common characteristics make it possible to combine all tasks into a single precedence diagram with I tasks. Tasks are classified into three categories under direction constraints: L-type tasks, which must be allocated to the left side; R-type tasks, which must be allocated to the right side; and E-type tasks, which can be assigned to either side. This study takes the following assumptions into account [23]:

- ✓ Energy consumption is computed using the power consumption of each robot, which is a presumption.
- ✓ According to the type of robot processing them, the operation times of tasks and the sequence-dependent setup times between two tasks only forward setup times in Scholl et al. [24] are taken into account in this paper—depend on the robots and are known and deterministic.
- ✓ Various models do similar tasks. Each model's processing time could be different or even zero.
- ✓ The number of available robots equals the number of workstations, and each robot is assigned to one of them.
- ✓ The times for material handling, loading, and unloading are not considered or already factored into the operating time of jobs.
- ✓ Work-in-process inventories and parallel workstations are not considered.

1.2 Problem formulation

The mathematical model used in this paper is taken from Aslan [22], with the addition of an objective function for energy consumption. The mathematical notation used in the formulations is provided below:

Indices

i,h,p,c task
j,g mated station
m product model
k,f side of the line

$k,f = \begin{cases} 1 \text{ indicates a left} - \text{side station} \\ 2 \text{ indicates a right} - \text{side station} \end{cases}$

(j,k) station of mated station *j* and its operation direction *k*

Parameters

I set of tasks in the combined precedence diagram; i= 1,..., I
J set of mated stations; j=1,..., J
M set of product models; m=1,..., M
R number of robot types; r=1,..., R
A_L set of tasks that should be performed at a left-side station; $A_L \subset$ I
A_R set of tasks that should be performed at a right-side station; $A_R \subset$ I
A_E set of tasks that can be performed on either side of a0station; $A_E \subset$ I
P(i) set of immediate predecessors of task *i*
$P_a(i)$ set of all predecessors of task *i*
S(i) set of immediate successors of task *i*
$S_a(i)$ set of all successors of task *i*
P_0 set of tasks that have no immediate predecessors; p_0={i ∈I| p(i)= φ}
t_{imr} completion time of task for model *m* by robot *r*
PC_r Operation power consumption of the robot *r* per time unit
D set of tasks and models with 0 processing times in all robots
φ a very large positive number
K(i) set indicating the preferred operation directions of task *i*

$$K(i)=\begin{Bmatrix} 1 & if & i \in A_R \\ 2 & if & i \in A_L \\ 1,2 & if & i \in A_E \end{Bmatrix}$$

S_{ipmr} setup time between two successive tasks *i* and *p* of model m with robot *r* if *i* and *p* are assigned to the same mated station

Decision Variables

c_m cycle time of model *m*
x_{ijk} 1 if task *i* is assigned to station *(j, k);* 0 otherwise
y_{rjk} 1 if robot *r* is assigned to station *(j, k);* 0 otherwise
z_{ip} 1 if task *i* is assigned before task *p* in the same station; 0 otherwise
w_{imjk} number of successors of task *i,* model *m*, mated station *j*, side *k*
ϕ_{imjk} 1 if task *i*, model *m* is the first of mated station *j*, side *k*
ω_{imjk} 1 if task *i*, model *m* is the last of mated station *j*, side *k*
t_{im}^f finish time of task *i* for model *m*

There are two objective functions in this paper. The objective function of time based model (Eq. 1) aims to minimize the cycle time. The objective function of the energy-based model (Eq. 2) aims to minimize the total energy consumption. Equation (1) is substituted by Equation (2) in the energy-based model. The mathematical model is given below:

$$\min \quad \sum_{m=1}^{M} c_m \tag{1}$$

$$min \quad \sum_{j=1}^{J}\sum_{k=1}^{2}\sum_{r=1}^{R} y_{rjk} \cdot PC_r \cdot \sum_{i=1}^{I}\sum_{m=1}^{M} t_{imr} \cdot x_{ijk} + \sum_{r=1}^{R}\sum_{j=1}^{J}\sum_{k=1}^{2} 0.1 \cdot y_{rjk} \cdot PC_r \cdot \left(\sum_{m=1}^{M} c_m - \sum_{m=1}^{M}\sum_{i=1}^{I} t_{imr} \cdot x_{ijk}\right) \tag{2}$$

Subject to:

$$\sum_{j=1}^{J}\sum_{k\in K(i)} x_{ijk} = 1 \qquad \forall i \in I \tag{3}$$

$$\sum_{g=1}^{J}\sum_{k\in K(i)} g \cdot x_{hgk} - \sum_{j=1}^{J}\sum_{k\in K(i)} j \cdot x_{ijk} \leq 0 \qquad \forall i \in \text{I-P}_0, \quad h \in P(i) \tag{4}$$

$$t_{im}^f \leq c_m \qquad \forall i \in \text{I}, \ \text{m} \in \text{M} \tag{5}$$

$$t_{im}^f - t_{hm}^f + \varphi\left(1 - \sum_{k\in K(h)} x_{hjk}\right) + \varphi\left(1 - \sum_{k\in K(i)} x_{ijk}\right) \geq \sum_{r=1}^{R} t_{imr} y_{rjk}$$
$$\forall i \in \text{I-P}_0, \text{h} \in P(i), \text{j} \in \text{J}, \ \text{m} \in M: (i,m) \notin D \tag{6}$$

$$t_{pm}^{f} - t_{im}^{f} + \varphi(1 - x_{pjk}) + \varphi(1 - x_{ijk}) + \varphi(1 - z_{ip}) \geq \sum_{r=1}^{R} (t_{pmr} + S_{ipmr}) y_{rjk}$$
$$\forall i \in I, \forall p \in I, \text{i}\neq\text{p}, m \in \text{M}, \ (i,m) \notin D, (p,m) \notin D \quad (7)$$

$$t_{im}^{f} - t_{pm}^{f} + \varphi(1 - x_{pjk}) + \varphi(1 - x_{ijk}) + \varphi(z_{ip}) \geq \sum_{r=1}^{R} (t_{imr} + S_{pimr}) y_{rjk}$$
$$\forall i \in I, \forall p \in I, \text{i}\neq\text{p}, m \in \text{M}, \ (i,m) \notin D, (p,m) \notin D \quad (8)$$

$$t_{im}^{f} + \varphi \cdot x_{ijk} \geq y_{rjk} \cdot t_{imr} \quad \forall i \in \text{I}, \ \text{m} \in \text{M}, \text{j} \in \text{J}, \text{r} \in \text{R}, (i,m) \notin D, \text{k} \in K(i) \quad (9)$$

$$\sum_{r=1}^{R} y_{rjk} = 1 \quad \forall j \in \text{J}, \ \text{k} \in \text{K}(i) \quad (10)$$

$$\sum_{j=1}^{J} \sum_{k=1}^{2} y_{rjk} = 1 \quad \text{r} \in \text{R} \quad (11)$$

$$\omega_{imjk} \geq x_{ijk} - w_{imjk} \quad \forall i \in I, m \in \text{M}, j \in \text{J}, \text{k} \in K(i): (i,m) \notin D \quad (12)$$

$$\phi_{imjk} \geq w_{imjk} + 1 - \sum_{p \in I} x_{pjk} \quad \forall i \in I, m \in \text{M}, j \in \text{J}, \text{k} \in K(i): (i,m) \notin D \quad (13)$$

$$\sum_{p \in I: i \neq p, (p,m) \notin D} z_{ip} \leq w_{imjk} + \varphi \cdot x_{ijk} \quad \forall i \in I, m \in \text{M}, j \in \text{J}, \text{k} \in K(i): (i,m) \notin D \quad (14)$$

$$\varphi \cdot x_{ijk} \geq w_{imjk} \quad \forall i \in I, m \in \text{M}, j \in \text{J}, \text{k} \in K(i): (i,m) \notin D \quad (15)$$

$$t_{im}^{f} + (\omega_{imjk} + \phi_{pmjk} - 1) \cdot S_{ipmr} \leq c_m + \varphi(1 - x_{ijk}) + \varphi(1 - y_{rjk}) \quad \forall i \in \text{I}, p \in \text{I}, \text{r} \in \text{R}, j \in \text{J}, \text{k} \in K(i) \cap K(p), \text{m} \in \text{M}: \text{i}\neq\text{p}, (i,m) \notin D, (p,m) \notin D \quad (16)$$

Constraint (3) ensures that each task is assigned to only one station. Constraint (4) enforces the precedence relationships among tasks. Constraint (5) guarantees that the finish time of each task for each product model in the set *M* does not exceed the cycle time. Constraint (6) ensures that a task *i* can only start once its immediate predecessor task h has finished. When tasks i and h are assigned to the same mated-station *j*, constraint (6) becomes $t_{im}^{f} - t_{hm}^{f} \geq \sum_{r=1}^{R} t_{imr} y_{rjk}$. For each pair of tasks assigned to the same station *(j, k)*, either constraint (7) or (8) becomes active. If task *i* is assigned earlier than task *p*, constraint (7) becomes $t_{pm}^{f} - t_{im}^{f} \geq \sum_{r=1}^{R} (t_{pmr} + S_{ipmr}) y_{rjk}$. On the other hand, if task *p* is assigned earlier than task *i*, constraint (8) becomes $t_{im}^{f} - t_{pm}^{f} \geq \sum_{r=1}^{R} (t_{imr} + S_{pimr}) y_{rjk}$. Constraint (9) guarantees that the finishing time of task *i* for a particular product model *m* is greater than or equal to the completion time of task *i* for all product models in the set *M*. Constraint (10) restricts the assignment of robots to stations such that only one robot can be assigned to a given station *(j, k)*. Constraint (11) ensures that each robot can only be assigned to a single station. Constraint (12) verifies whether task *i* is the final task to be executed in a station, in the scenario where there are no successors to this task. Constraint (13) validates whether task *i* is the first task to be executed in a specific station when all the other assigned tasks are successors of task *i*. Constraint (14) ensures that the total number of successors that are executed in the same station for a given task corresponds to the number of tasks that are executed immediately after that task. Constraint (15) guarantees that a task has successors only if it is executed in the specified station. Lastly, constraint (16) verifies that the cycle time of the model is equivalent to the total setup time between the last task of the station and the first task.

In this study, the power consumption of a robot is divided into two clear-cut categories: production mode and standby mode. Using PC_rx t_{imr} it is possible to calculate the energy consumptions of robots for production mode. The energy consumed during the robot's active operation is referred to as the production mode, as explained by the first portion of Equation (2). The second part of Equation (2) describes the energy consumed during the times between operations, which is interchangeable with the term "standby mode," and is referred to as the "standby power consumption". 10% of the robot's operation energy is used in standby mode each time unit [16]. The standby time formula (Eq. 17) is as follows:

Standby time of the workstation= Cycle time of the assembly line – total process time of the workstation *(17)*

2 Proposed Hybrid Genetic Algorithm

In this paper a new hybrid genetic algorithm (hGA) with adaptive local search scheme is proposed. The local search scheme is based on Yun [25]. At each generation the local search is applied based on the fitness value ratio, calculated as Equation (18):

$$\text{FVR(t)} = \frac{\overline{\text{f}_{\text{newpop(t)}}}}{\text{f}_{\text{newpop(t-1)}}} \quad (18)$$

Where $\overline{\text{f}_{\text{newpop(t)}}}$ signifies the mean fitness value of the freshly generated population resulting from the selection strategy that utilizes both the parent and offspring populations in generation t. The local search is applied if FVR(t) > 1, which means that algorithm is not converging and application of an adaptative local search is required.

The key elements of proposed hGA are:

- Initial population generation
- Crossover
- Mutation
- Fitness evaluation and selection
- Local search

2.1 Initial population generation

The first step of hGA is generating an initial population, that is a set of individuals. Each individual is a solution to the problem. Each one has a set of genes that form a chromosome. Solution representation of a MRTABLPS, which is depicted in Fig. 2, represented by four vectors [22]:

"Robot assignment vector (R) shows the robots assigned to each station. R is number of robots, and it is a (1xR) vector.

Task sequence vector (T) indicates task sequence according to precedence constraints. It is a (1xI) vector, where I is the number of assembly task.

Side assignment vector (S), represents the side information of each task. It can be 'l' (for left) or 'r' (for right), if task i ∈ A_E, it

is randomly assigned to the left side or right side. S vector is a (1xI) vector, where I is the number of assembly task.

Breakpoint vector (B) is used to show the position in which a mated station stops." (pp. 998-999).

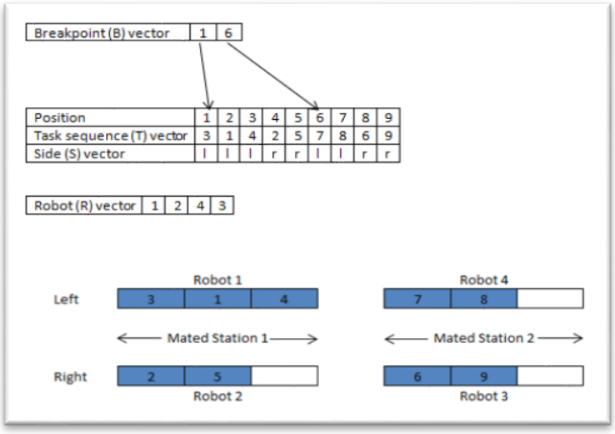


Figure 2. Solution representation of hGA

The process for creating the initial population is described below to ensure sufficient variation. Firstly, the jobs with no preceding tasks are assigned. Then a task is randomly chosen and added to the first position of the T vector. The immediate successors of the selected task are added to the available task list. This process is repeated until there are no remaining tasks. It should be noted that the hGA algorithm only considers solutions within the defined search space, meaning that infeasible solutions are eliminated and that the precedence constraints in the T vector must be followed. If a task has a preferred side, it is allocated to that side; otherwise, a side is selected at random. The breakpoint vector and robot assignments are filled in a random manner.

2.2 Crossover

In this study several kinds of crossover operators are implemented. OX operator (shown in Fig. 3) which was proposed by Goldberg [26] is made for problems that are order based and it works as following: given the T vectors of each parent and a random selected position, first child inherits the left side (from start to the random chosen point) from a parent, and the rest is filled by the missing elements in the child in the order they appear in the other parent. The side vector (S) is also recombined accordingly.

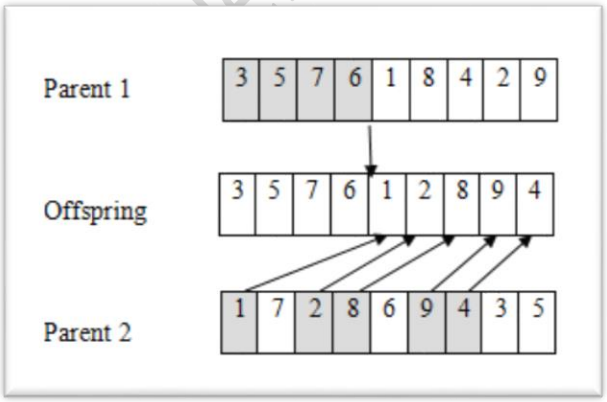


Figure 3. Illustration of the OX operator

R vector uses the Partially Mapped Crossover (PMX) [27]. as crossover operator This crossover operator chooses two genes (known as cut points) from each parent at random first. The genetic material from the second parent is replaced in order to construct the genetic material between the two cut points in the first parent. Then, outside of the cut points, the inverse replacement is used. As an illustration, the tours of the two parents with a random cut point between the second and third string element and another random cut point between the fourth and fifth string element are as follows:

Parent 1 – [2 3 | 5 6 | 1 4]

Parent 2 – [6 2 | 4 1 | 3 5]

The mapping sections are the substrings that lie between the cut points. The mappings in our example are specified as 5 ↔ 4 and 6 ↔ 1. The second parent's mapping part is duplicated into the first offspring, while the first parent's mapping section is copied into the second offspring in this order.

Offspring 1 – [x x | 4 1 | x x]

Offspring 2 – [x x | 5 6 | x x]

The elements of the i-th parent are then copied into offspring i (i = 1, 2) to finish it out. If a string is found in the offspring already, it is replaced in accordance with the mappings.

Offspring 1 – [2 3 | 4 1 | 6 5]

We similarly get

Offspring 2 – [1 2 | 5 6 | 3 4]

Two types of crossovers are used for the B vector. At crossover phase they are chosen randomly.

- The children get all the genes related to R, T and S vector to a parent and B from the other parent
- Arithmetic Recombination: the two parents are combined here following an arithmetic recombination scheme. Given the two breakpoint vectors the children genes are the average values of the parents, rounded to the nearest integer. For example:

Parent 1 – [1,5,9]

Parent 2 – [1,3,8]

Children will be

(1+1)/2 = 1, (5+3)/2 = 4, (9+8)/2 = 8.5 (rounds to 9).

Child – [1,4,9]

In hGA two parents produce two children. It is also assumed that all children are feasible considering the problem constraints. For all individuals chosen to mate, the operators below are used or not with 50% probability. Detail working procedure of the crossover operators is presented below:

Algorithm 1. Crossover

Input: crossover_prob (crossover probability) and two parents (parent1 and parent2)

r ← random(0,1)

if r < crossover_prob

r2 ← random(0,1)

if r2 < 0.5

op ← random(0,1)

if op < 0.5

child1, child2 ← Arithmetic Recombination(parent1.B_vector, parent2. B_vector)

```
        else
                child1 ← parent2.B_vector
                child2 ← parent1.B_vector
    r2 ← random(0,1)
    if r2 < 0.5
        child1. T_vector, child2. T_vector ← OX (parent1. T_vector, parent2. T_vector)
    if r2 < 0.5
        child1. R_vector, child2. R_vector ← PMX (parent1. R_vector, parent2. R_vector)
  endif
Output: two children
```

2.3 Mutation

The mutation is the operator that guarantees diversification in the population. The mutation, typically a random process, involves altering a solution and is directed towards moving away from local optima. Mutation size is a parameter of the mutation function. Mutation can incite in none, one, two, three or four structures of the solution (robot assignment, tasks order assignment, side assignment and mated station assignment). The operators that will be used is based on a generated uniform number inside mutation function.

Algorithm 2. Mutation

```
Input: mutation_prob (mutation probability) and solution
  size_mutation1 ← random(1, mutation_prob*num_stations)
  size_mutation2 ← random(1, mutation_prob*num_tasks)
  size_mutation3 ← random(1, mutation_prob*num_tasks)
  size_mutation4 ← random(1, mutation_prob*num_mated_stations)
  r ← random(0,1)
  if r < 0.5
      r2 ← random(0,1)
      if r2 < 0.5
          solution ← swap_robots(solution, size_mutation1)
      else
      solution ← insert_robots(solution, size_mutation1)
      endif
  endif
  r ← random(0,1)
  if r < 0.5
      r2 ← random(0,1)
      if r2 < 0.3
          solution ← insert_task_random(solution, size_mutation2)
      elseif r2 < 0.6
          solution ← insert_task_last (solution, size_mutation2)
      else
          solution ← insert_task_first(solution, size_mutation2)
      endif
  endif
  r ← random(0,1)
  if r < 0.5
      solution ← swap_side(solution, size_mutation3)
  r ← random(0,1)
  if r < 0.5
      solution ← move_breakpoint_vector(solution, size_mutation4)
Output: Mutation solution
```

2.3.1 Robot assignment

If robot assignment structure (R vector) is called, there are two types of possible mutation movements:

•Swap: choose two stations randomly, and swap their robot assignment

•Inserts: choose two stations randomly, and insert the first chosen robot in the position of the second

The number of robots swaps or insertions are based on a random generated number.

2.3.2 Task order assignment

If tasks order structure (T vector) is called, a task is selected at random, and its evaluation involves determining how many positions it can move both to the left and right without reaching an immediate predecessor or successor. There are three types of possible mutation movements.

•choose one position among those ones and insert the task in it

•insert the task in the last possible position

•insert the task in the first possible position

The number of task insertions are based on a random generated number.

2.3.3 Side assignment

If side assignment structure (S vector) is called, a task that does not have preferred side at random is chosen and it is assigned to the opposite side. The number of station reassignments are based on a random generated number.

2.3.4 Mated station assignment

If mated station assignment structure (B vector) is called, one position is randomly selected in the breakpoint vector. The assessment involves determining the number of shifts it can make to the left or right without reaching the adjacent or final station. Among these potential shift values, one is selected to replace the current value. The number of mated station reassignments are based on a random generated number. A general schema of mutation operator is shown below:

2.4 Fitness evaluation and selection

The fitness value of an individual is the value of the objective function that is the sum of cycle times or total energy consumptions for all models. Selection is the process of selecting parents to breed their genes to produce child for next generation. In this study, it is used the binary tournament selection where two individuals at random are chosen, and the one with better (lower cycle time or lower total energy consumption) fitness will be selected to breed.

2.5 Local search

As mentioned in Section 3, it is recommended to integrate a local search method into the genetic algorithm loop. This will enhance the algorithm's ability to converge towards the optimal solution if the average fitness value in the current generation is greater than that of the previous one. This situation indicates that generations t and t-1 are not converging towards the optimal solution.

This study makes use of the local search algorithm created by Aslan [22] for MRTALBPS-II. All of the following will be carried out during the local search phase until neither of them is able to come up with a better solution. When a move is made, only the portion that was affected by the move needs to be calculated

from scratch because each neighbourhood only deals with one portion of the solution.

2.5.1 Robot assignment

For time-based models, the station with the longest processing time is chosen for each model, and all the other robots are allocated to this station. Afterward, the robots that can achieve the shortest finish time are selected.

In the case of energy-based model, the station with maximum total energy consumption for each model is selected and all other robots are assigned to this station. Then the robots providing the lowest total energy consumption is selected. During each iteration of the local search, only the two stations being considered for a swap need to be re-evaluated, as the statuses of the other stations remain unchanged.

2.5.2 Task assignment

In the context of the time-based model, the approach involves selecting the station with the longest completion time for each model. Then, for each job assigned to this station, starting with the first job, we examine all possible insertions in subsequent positions that do not violate precedence constraints. We choose the best task and insert it in the optimal location to minimize the overall finish time.

In contrast, when using the energy-based model, the station that consumes the most energy across all models is picked. Similarly, for each job assigned to this station, beginning with the first job, we explore potential insertions in subsequent positions while ensuring precedence constraints are met. The goal is to find the best task and insert it in the optimal location to minimize overall energy consumption. It's important to note that only the worst-performing station needs to be reevaluated in each iteration of this local search because the others remain unchanged.

2.5.3 Side assignment

In the time-based model, the station with the longest finish time for each model is selected. Tasks without a favored direction assigned to this station are reversed, and the best assignment that minimizes the finish time is chosen.

In the energy-based model, for each model, the station with the highest overall energy consumption is designated. Tasks without a favored direction assigned to this station are reversed as well, and the best assignment that minimizes the overall energy consumption is selected. Notably, only the least efficient station needs reevaluation in each iteration of this local search, as the others remain unchanged.

2.5.4 Mated-station assignment

In the time-based model, the station with the longest finish time for each model is chosen. The examination involves considering the option of moving the assigned matching station to both the left and right directions, and the breakpoint vector can be employed for this purpose.

In the energy-based model, for each model, the station with the highest overall energy consumption is selected. The assessment includes exploring the possibility of relocating the assigned matching station in both left and right directions, with the aid of the breakpoint vector.

The proposed hybrid genetic algorithm (hGA) is presented below in broad form:

Algorithm 3. hGA algorithm

Input: Algorithm parameters (**population_size, mutation_prob, crossover_prob, local_search_prob)**
Population ← []
for k in range(0, population_size):
S ← Generate Initial Solution
Population ← Population + S
endfor
while(time_elapsed <= (num_tasks*num_tasks*60)/1000):
for k in range(0, population_size):
parent1 ← Get random solution from population
parent2 ← Get random solution from population
child1, child2 ← crossover(parent1, parent2, crossover_prob)
Population ← Population + child1
Population ← Population + child2
Endfor
Calculate FVR(t)
for k in range(0, population_size):
solution1 ← Get random solution from population
solution1' ← mutation(solution1, mutation_prob)
if FVR(t) > 1
solution1' ← local_search(solution1', local_search_prob)
Population ← Population + solution1'
endfor
Population ← Select population_size best elements from Population
endwhile
Output: Total cycle time and total energy consumption of models and the detailed task and robot assignment

3 Computational Experiments

This section clarifies the methodology employed to evaluate the efficacy of hGA in addressing time and energy-based models. To evaluate the effectiveness of the hGA, a series of test problems were examined. These problems consisted of four small-sized problems (P9, P12, P16, and P24) and three large-sized problems (P65, P148, and P205). The issue numbers P9, P12, and P24 have been sourced from Kim et al. [28], while P16, P65, and P205 have been sourced from Lee et al. [29] Additionally, problem P148 has been taken from Bartholdi [8]. The study directions and priority diagrams utilized in the present study are derived directly from academic literature.
The operation time for task *i* was generated randomly by Robot R within the range of $[t_i \times 0.8, t_i \times 1{,}2]$, and t_i represented the original operation time in the study conducted by Özcan and Toklu [30] and Delice et. al. [31]. Sequence-dependent setup times' random matrix was constructed by employing a uniform distribution of U[0,0.75*$min_{\forall i \in N} t_i$]. All models have the same overall unit number ratio ($q_A = q_B = \ldots = q_m$). The Robot R's standby energy consumption was found to be 10% of its operational energy usage, as reported by Li et al. [17]. The energy consumption of robots during operation, measured in units of time, is provided in Appendix.

Following plenty of initial experiments, the parameters for the hGA are established as given in Table 2:

Table 2. Parameters selected for the hGA

Algorithm	Parameters	Range	Selected Value
hGA	Population size	50, 100, 200	100
	Crossover probability	0.25, 0.5, 0.75	0.75
	Mutation probability	0.1, 0.2, 0.3	0.1
	Apply local search	True, False	True
	Local Search Probability	0.1,0.2,0.3	0.2

The control parameters utilized for VNS are obtained from the publication of Aslan [22].

3.1 Comparison between hGA and MIP

As it can be seen from Table 3, CPLEX could achieve 6 optimal solutions out of 9 instances of P9, P12, P16 and P24 in the case of time-based model. Furthermore, CPLEX gets 5 optimal solutions out of 9 instances of P9, P12, P16 and P24 in the case of energy-based model. This situation demonstrates the complexity of the considered problem. On the other hand, hGA reaches the optimal solutions for every instance of small-sized test problems that MIP found optimal solutions for both time and energy based models in short CPU times.

Table 3. Optimal solutions for small problems obtained using CPLEX solver

		CPLEX Solver								hGA				
		Time based model				Energy based model				Time based model		Energy based model		
Problem	Nm	CT	GAP (%)	TEC	CPU time(s)	CT	TEC	GAP (%)	CPU time(s)	CT	TEC	CT	TEC	CPU time(s)
P-9	2	8.6		8.0	1.42	10.6	7.5		2.34	8.6	8.0	10.6	7.5	0.81
P-9	3	6.7		9.7	3.25	7.9	8.2		411.69	6.7	9.7	7.9	8.2	0.81
P-12	2	12.7		15.5	10.84	18.2	12.9		64.59	12.7	15.5	18.2	12.9	1.44
P-12	3	9.8		14.1	68.66	10.2	12.8		5,449.26	9.8	14.1	10.2	12.8	1.44
P-16	2	39.6		44.5	115.97	51.6	38.7		635.44	39.6	44.5	51.6	38.7	2.56
P-16	3	28		51.5	349.00	41.2	43.2	20	>7,200	28	51.5	40.3	44.0	2.56
P-24	2	68.4	9.60	187.0	>7,200	111.6	179.0	18.73	>7,200	69.1	187.4	111.8	179.0	5.76
P-24	3	46.7	23.63	174.8	>7,200	97	152.8	46.30	>7,200	48.8	174.6	86.0	151.6	5.76
P-24	4	38.1	30.71	208.6	>7,200	60.7	164.4	70.33	>7,200	34.8	188.4	58.2	162.9	5.76

Nm: number of mated stations; CT: cycle time; TEC: Total energy consumption; CPU: central processing unit

3.2 Comparison between hGA and VNS

As CPLEX is incapable of solving large-sized problems, it is conducted another experiment on large-sized problems (P65, P148, and P205) to compare the performance of hGA and variable neigborhood search algorithm (VNS) which was proposed by Aslan [22] for MRTALBPS-II. One might see that the time required to compute the time-based model is significantly lower than the time required to compute the energy-based model from Table 3. Therefore, the CPU time for large problems experiments is set to be NT × NT × 30 ms for time based model and NT × NT × 60 ms for energy based model.

In the experiment, each algorithm was applied to solve the given cases ten times, and the outcomes were documented for each run. The processing times were then converted to relative percentage deviations (RPD) based on a formula (Eq. 19), which compares the cycle time or total energy consumption of one combination Fit_{some} to the best cycle time or best total energy consumption of all combinations Fit_{best}.

$$\text{RPD} = 100 \times (Fit_{some} - Fit_{best}) / 100 \quad (19)$$

Table 4 shows the average RPD values for large-sized instances in ten repetitions for two algorithms that were tested. Table 4 demonstrates that the hGA outperformed the VNS algorithm for both time-based and energy-based models. In order to assess the statistical significance of the observed variances, a non-parametric Mann-Whitney U test is utilized due to the violation of normality in the residuals. Due to significant fluctuations in algorithm performance across various instances, this study adopts the average Relative Percentage Deviation (RPD) of all instances within a single run as the chosen response variable. Table 5 shows that the p-values in two categories are significantly less than 0.05, indicating the presence of significant differences in the solution qualities between hGA and the VNS. This confirms the proposed hGA's superiority from a statistical perspective.

Table 4. Average RPD values of compared algorithms

Problem	Nm	Time based model			Energy based model		
		VNS	hGA	CPU time(s)	VNS	hGA	CPU time(s)
P-65	4	5.31	3.35	126.75	5.79	4.51	253.5
P-65	5	9.65	9.36	126.75	11.65	4.86	253.5
P-65	6	8.76	2.85	126.75	11.61	10.61	253.5
P-65	7	9	5.32	126.75	5.56	7.73	253.5
P-65	8	2.41	4.97	126.75	6.8	7.81	253.5
P-148	4	5.32	3.34	657.12	5.26	2.35	1,314.24
P-148	5	4.04	2.9	657.12	8.68	6.04	1,314.24
P-148	6	9.2	1.59	657.12	21.16	7.37	1,314.24
P-148	7	11.74	9.43	657.12	19.99	8.28	1,314.24
P-148	8	16.37	11.18	657.12	12.48	9.11	1,314.24
P-148	9	15.23	19.3	657.12	27.68	15.02	1,314.24
P-148	10	17.94	19.07	657.12	17.84	14.8	1,314.24
P-148	11	7.86	0.5	657.12	11.45	7.79	1,314.24
P-148	12	11	9.77	657.12	17.69	9.16	1,314.24
P-205	4	6.12	0.48	1260.75	4.25	3.73	2521.5
P-205	5	4.99	2.04	1260.75	18.02	4.81	2521.5
P-205	6	19.85	11.48	1260.75	5.01	7.64	2521.5
P-205	7	14.11	6.35	1260.75	9.49	1.89	2521.5
P-205	8	14.96	16.94	1260.75	26.63	13.18	2521.5
P-205	9	17.71	7.47	1260.75	32.68	11.27	2521.5
P-205	10	22.02	7.93	1260.75	5.94	9.18	2521.5
P-205	11	13.95	12.18	1260.75	16.71	8.19	2521.5
P-205	12	10.26	8.42	1260.75	15.02	18.99	2521.5
P-205	13	21.34	13.87	1260.75	15.04	19.69	2521.5
P-205	14	12.57	4.46	1260.75	7.63	8.71	2521.5
Average RPD		**11.67**	**7.78**		**13.60**	**8.91**	

Table 5. Outcomes of Mann-Whitney U test between hGA and VNS

Mann-Whitney U	188.500
Wilcoxon W	513.500
Z	-2.406
Asymp. Sig. (2-tailed)	.016

a)Time-based model outcome

Mann-Whitney U	20.250
Wilcoxon W	527.500
Z	-2.134
Asymp. Sig. (2-tailed)	.033

b) Energy-based model outcome

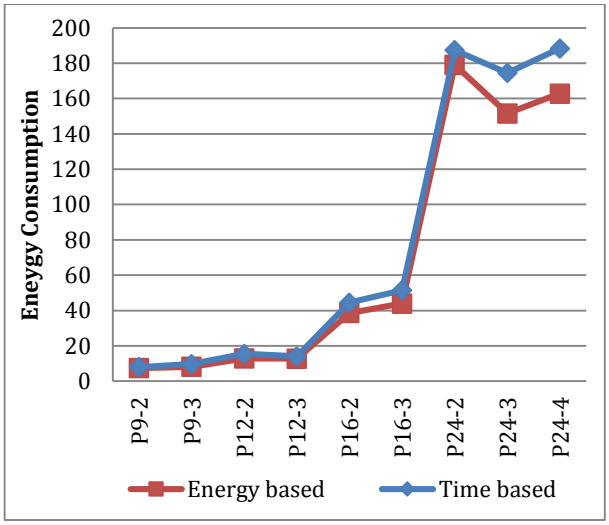
3.3 Performance of two models proposed

The time-based and energy-based models are assessed based on cycle time and total energy consumption, with Tables 3 and 6 presenting the results for small and large problems, respectively. Table 3 and 6 present the results produced by hGA those are the best cycle times and total energy consumptions throughout the course of 5 runs.

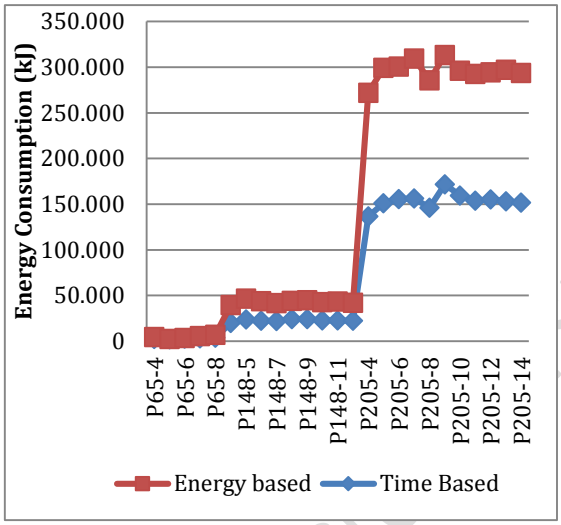
Table 6. Total energy consumption and cycle time for large datasets

		hGA					
Problem	**Nm**	**Time based model**			**Energy based model**		
		CT	TEC	CPU time	CT	TEC	CPU time
P-65	4	1024.8	2,370.4	126.75	1,541.3	2,100.0	253.5
P-65	5	851.5	2,146.7	126.75	1,507.7	493.8	253.5
P-65	6	729.4	2,717.6	126.75	1,530.5	703.4	253.5
P-65	7	665.6	3,209.9	126.75	1,549.1	2,363.4	253.5
P-65	8	584.1	4,027.2	126.75	1,537.7	2,903.1	253.5
P-148	4	1964.1	20,003.2	657.12	2,415.2	19,529.2	1,314.24
P-148	5	1823.8	23,757.0	657.12	1,874.7	22,980.8	1,314.24
P-148	6	1015.0	22,515.3	657.12	1,108.2	21,274.7	1,314.24
P-148	7	1005.8	22,007.6	657.12	1,319.9	19,618.0	1,314.24
P-148	8	841.5	24,230.2	657.12	1,083.1	20,072.3	1,314.24
P-148	9	850.2	24,355.2	657.12	920.5	20,401.1	1,314.24
P-148	10	710.0	22,728.2	657.12	946.4	19,961.0	1,314.24
P-148	11	684.7	22,943.2	657.12	766.1	20,592.0	1,314.24
P-148	12	588.3	22,424.2	657.12	712.4	19,667.4	1,314.24
P-205	4	11,936.0	136,593.6	1260.75	13,878.6	135,465.0	2521.5
P-205	5	11,034.4	151,199.7	1260.75	13,720.6	148,064.7	2521.5
P-205	6	6,103.8	155,496.3	1260.75	8,853.1	145,214.3	2521.5
P-205	7	5,282.1	156,402.5	1260.75	8,461.8	153,083.9	2521.5
P-205	8	4,688.2	146,052.8	1260.75	4,879.1	139,458.2	2521.5
P-205	9	4,198.5	171,601.5	1260.75	5,333.0	141,842.7	2521.5
P-205	10	4,152.3	159,528.2	1260.75	5,384.7	136,470.6	2521.5
P-205	11	3,620.5	153,962.5	1260.75	4,342.4	138,707.0	2521.5
P-205	12	3,360.9	155,317.9	1260.75	4,671.9	139,209.5	2521.5
P-205	13	3,062.4	153,116.7	1260.75	3,092.5	144,065.4	2521.5
P-205	14	2,940.7	151,940.0	1260.75	4,049.5	141,615.8	2521.5

The average energy consumption of time-based model is 77 kJ for small problems and 76,426 kJ for large problems. On the other hand, average energy consumption of energy-based model is 69 kJ for small problems and 70,234 kJ for large problems. The average energy savings is found to be 8 kJ for small problems and 6,192 kJ for the large problems. It is observed from Fig. 6 and 7 that when task number increases energy saving also increases. Furthermore, the average cycle time of time-based model is 29 units for small problems and 2,949 units for large problems, whereas the average cycle time of energy-based model is 44 units for small problems and 3,819 units for large problems. The average cycle time reduction is found to be 15 units for small problems and 870 units for large problems. Fig. 8 shows the cycle time comparison between time based and energy based models for small and large problems.

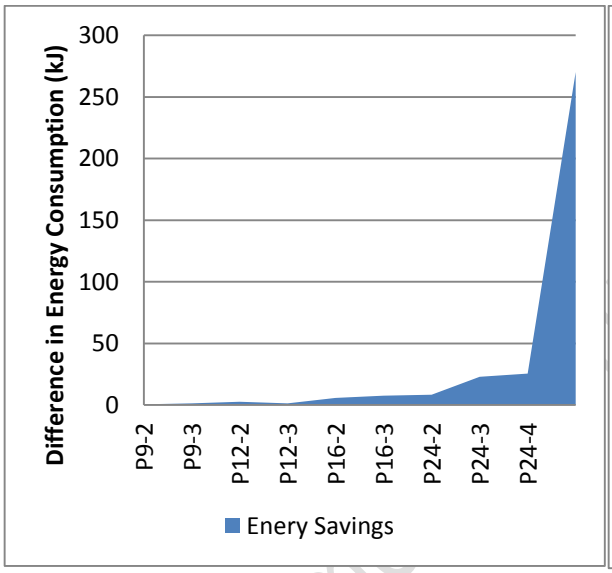


a

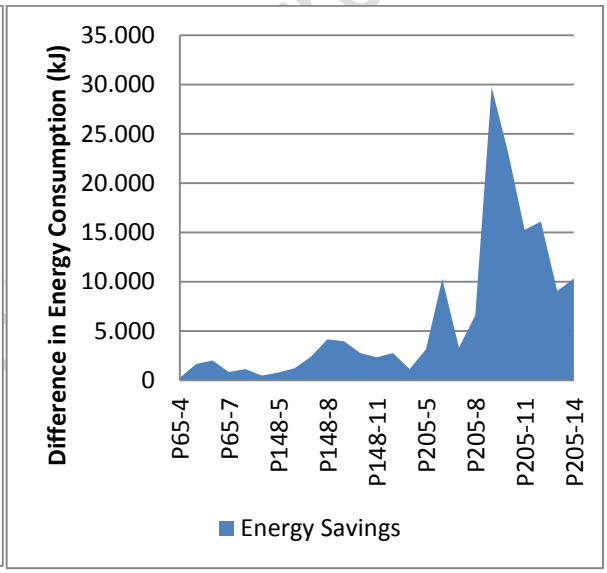


b

Fig. 6. a) Comparision of the energy usage of two models in small problems
b) Comparision of the energy usage of two models in large problems

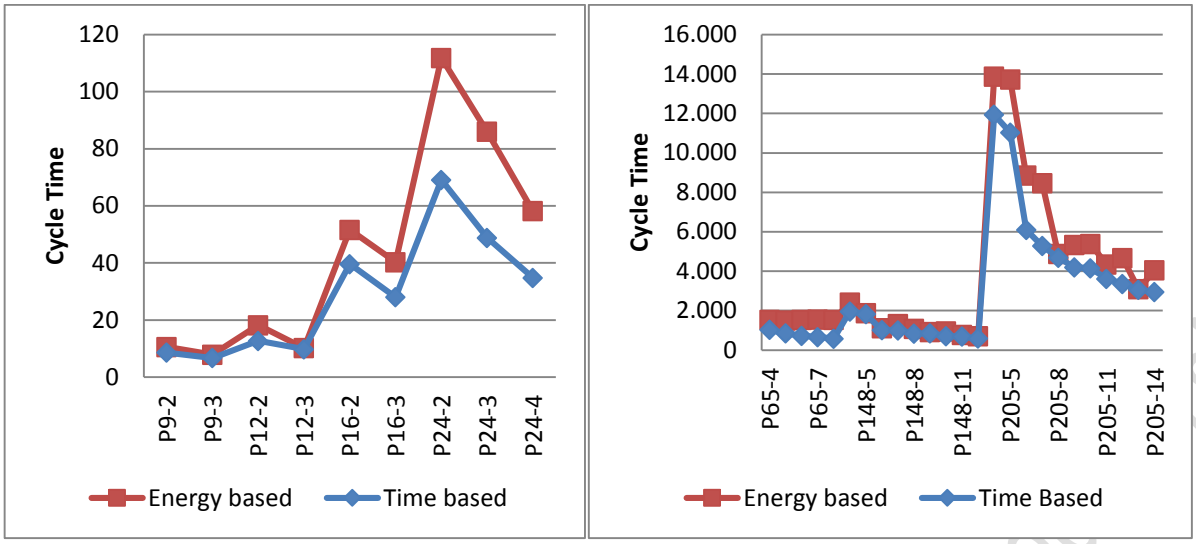


a



b

Fig. 7. a) The possible energy savings for the energy-based model in small problems
b) The possible energy savings for the energy-based model in large problems



a b

Fig. 8. a) Comparision of the cycle time of two models in small problems
b) Comparision of the cycle time of two models in large problems

4 Conclusion

Today's globalized and competitive world requires organizations to have an ever-increasing performance to remain existing. Energy and energy costs for manufacturers are an important key to global competition. Due to the rising energy costs, manufacturing industries place significant emphasis on reducing energy consumption and promoting an environmentally sustainable workplace. Therefore, optimizing consumption of energy and cycle time is an important issue for manufacturing systems.

This paper enhances Aslan's [22] mixed-integer programming model for MRTALBPS-II by introducing a new objective function to calculate total energy consumption. It proposes an effective hybrid genetic algorithm (hGA) with an adaptive local search to solve large instances efficiently. Unlike prior studies focused on cycle time and cost, this research aims to simultaneously minimize cycle time and total energy usage in robotic assembly line balancing. A comparative analysis reveals that the proposed hGA outperforms the VNS algorithm from Aslan [22]. Furthermore, this study presents models with a dual focus on time and energy, essential factors in manufacturing operations, aimed at enhancing efficiency and minimizing energy consumption. The emphasis between time and energy may shift across various time periods based on management's priorities. Accordingly, the suitable model can be chosen based on the priority established by management.

In future works, energy efficient MRTALBPS-II studied in this paper can be solved by multi-objective evolutionary algorithms. Furthermore, the problem presented in this paper exclusively addresses robotic assembly lines. Nevertheless, there is a growing interest among manufacturing enterprises in human-robot collaboration. Therefore, future research endeavors could explore more realistic scenarios, incorporating aspects such as the interaction between human workers and robots.

5 Author contribution statements

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

6 Ethics committee approval and conflict of interest declaration

There is no need to obtain ethics committee permission for the article prepared.

There is no conflict of interest with any person / institution in the article prepared.

7 References

[1] "Key aspects of the Paris Agreement | UNFCCC." Accessed: Sep. 26, 2023. [Online]. Available: https://unfccc.int/most-requested/key-aspects-of-the-paris-agreement

[2] B. Sun, L. Wang, and Z. Peng, "Bound-guided hybrid estimation of distribution algorithm for energy-efficient robotic assembly line balancing," *Comput. Ind. Eng.*, vol. 146, p. 106604, Aug. 2020, doi: 10.1016/j.cie.2020.106604.

[3] A. Scholl, "Balancing and sequencing of assembly lines," Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), Publications of Darmstadt Technical University, Institute for Business Studies (BWL), Jan. 1999. Accessed: Sep. 26, 2023. [Online]. Available: https://econpapers.repec.org/paper/darwpaper/10881.htm

[4] N. Boysen, M. Fliedner, and A. Scholl, "Sequencing mixed-model assembly lines: Survey, classification and model critique," *Eur. J. Oper. Res.*, vol. 192, no. 2, pp. 349–373, Jan. 2009, doi: 10.1016/j.ejor.2007.09.013.

[5] P. Sivasankaran and P. Shahabudeen, "Literature review of assembly line balancing problems," *Int. J. Adv. Manuf. Technol.*, vol. 73, no. 9–12, pp. 1665–1694, Aug. 2014, doi: 10.1007/s00170-014-5944-y.

[6] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 694–715, Feb. 2006, doi: 10.1016/j.ejor.2004.07.023.

[7] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," *Eur. J. Oper. Res.*, vol. 183, no. 2, pp. 674–693, Dec. 2007, doi: 10.1016/j.ejor.2006.10.010.

[8] J. J. Bartholdi, "Balancing two-sided assembly lines: a case study," *Int. J. Prod. Res.*, vol. 31, no. 10, pp. 2447–2461, Oct. 1993, doi: 10.1080/00207549308956868.

[9] Z. Li, I. Kucukkoc, and J. M. Nilakantan, "Comprehensive review and evaluation of heuristics and meta-heuristics for two-sided assembly line balancing problem," *Comput. Oper. Res.*, vol. 84, pp. 146–161, Aug. 2017, doi: 10.1016/j.cor.2017.03.002.

[10] H. Soysal-Kurt and S. K. İşleyen, "Multi-objective optimization of cycle time and energy consumption in parallel robotic assembly lines using a discrete firefly algorithm," *Eng. Comput.*, vol. 39, no. 6, pp. 2424–2448, 2022, doi: https://doi.org/10.1108/EC-12-2020-0747.

[11] B. Zhou and Q. Wu, "Decomposition-based bi-objective optimization for sustainable robotic assembly line balancing problems," *J. Manuf. Syst.*, vol. 55, pp. 30–43, Apr. 2020, doi: 10.1016/j.jmsy.2020.02.005.

[12] J. Rubinovitz, J. Bukchin, and E. Lenz, "RALB – A Heuristic Algorithm for Design and Balancing of Robotic Assembly Lines," *CIRP Ann.*, vol. 42, no. 1, pp. 497–500, 1993, doi: 10.1016/S0007-8506(07)62494-9.

[13] M. Aghajani, R. Ghodsi, and B. Javadi, "Balancing of robotic mixed-model two-sided assembly line with robot setup times," *Int. J. Adv. Manuf. Technol.*, vol. 74, no. 5–8, pp. 1005–1016, Sep. 2014, doi: 10.1007/s00170-014-5945-x.

[14] Z. A. Çil, S. Mete, and K. Ağpak, "Analysis of the type II robotic mixed-model assembly line balancing problem," *Eng. Optim.*, vol. 49, no. 6, pp. 990–1009, Jun. 2017, doi: 10.1080/0305215X.2016.1230208.

[15] P. Chutima, "A comprehensive review of robotic assembly line balancing problem," *J. Intell. Manuf.*, vol. 33, no. 1, pp. 1–34, Jan. 2022, doi: 10.1007/s10845-020-01641-7.

[16] J. Mukund Nilakantan, G. Q. Huang, and S. G. Ponnambalam, "An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems," *J. Clean. Prod.*, vol. 90, pp. 311–325, Mar. 2015, doi: 10.1016/j.jclepro.2014.11.041.

[17] Z. Li, Q. Tang, and L. Zhang, "Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm," *J. Clean. Prod.*, vol. 135, pp. 508–522, Nov. 2016, doi: 10.1016/j.jclepro.2016.06.131.

[18] J. M. Nilakantan, Z. Li, Q. Tang, and P. Nielsen, "Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems," *J. Clean. Prod.*, vol. 156, pp. 124–136, Jul. 2017, doi: 10.1016/j.jclepro.2017.04.032.

[19] Z. Zhang, Q. Tang, and L. Zhang, "Mathematical model and grey wolf optimization for low-carbon and low-noise U-shaped robotic assembly line balancing problem," *J. Clean. Prod.*, vol. 215, pp. 744–756, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.030.

[20] Z. Zhang, Q. Tang, Z. Li, and L. Zhang, "Modelling and optimisation of energy-efficient U-shaped robotic assembly line balancing problems," *Int. J. Prod. Res.*, vol. 57, no. 17, pp. 5520–5537, Sep. 2019, doi: 10.1080/00207543.2018.1530479.

[21] İ. Baş, Ö. Tosun, and V. Bayram, "Line balancing optimization under robot location and worker-station assignment considerations: A case study of a dishwasher factory," *Pamukkale Univ. J. Eng. Sci.*, vol. 27, no. 4, pp. 495–503, 2021, doi: 10.5505/pajes.2021.48961.

[22] Ş. Aslan, "Mathematical model and a variable neighborhood search algorithm for mixed-model robotic two-sided assembly line balancing problems with sequence-dependent setup times," *Optim. Eng.*, vol. 24, no. 2, pp. 989–1016, Jun. 2023, doi: 10.1007/s11081-022-09718-3.

[23] Z. Li, M. N. Janardhanan, Q. Tang, and S. G. Ponnambalam, "Model and metaheuristics for robotic two-sided assembly line balancing problems with setup times," *Swarm Evol. Comput.*, vol. 50, p. 100567, Nov. 2019, doi: 10.1016/j.swevo.2019.100567.

[24] A. Scholl, N. Boysen, and M. Fliedner, "The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics," *Spectr.*, vol. 35, no. 1, pp. 291–320, Jan. 2013, doi: 10.1007/s00291-011-0265-0.

[25] Y. Yun, "Hybrid genetic algorithm with adaptive local search scheme," *Comput. Ind. Eng.*, vol. 51, no. 1, pp. 128–141, Sep. 2006, doi: 10.1016/j.cie.2006.07.005.

[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[27] Goldberg, D. E. and Lingle, R., "Alleles, Loci, and the Traveling Salesman Problem," in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, Psychology Press, 1985.

[28] Y. K. Kim, Y. Kim, and Y. J. Kim, "Two-sided assembly line balancing: A genetic algorithm approach," *Prod. Plan. Control*, vol. 11, no. 1, pp. 44–53, Jan. 2000, doi: 10.1080/095372800232478.

[29] T. O. Lee, Y. Kim, and Y. K. Kim, "Two-sided assembly line balancing to maximize work relatedness and slackness," *Comput. Ind. Eng.*, vol. 40, no. 3, pp. 273–292, Jul. 2001, doi: 10.1016/S0360-8352(01)00029-8.

[30] U. Özcan and B. Toklu, "Balancing of mixed-model two-sided assembly lines," *Comput. Ind. Eng.*, vol. 57, no. 1, pp. 217–227, Aug. 2009, doi: 10.1016/j.cie.2008.11.012.

[31] Y. Delice, E. Kızılkaya Aydoğan, U. Özcan, and M. S. İlkay, "A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing," *J. Intell. Manuf.*, vol. 28, no. 1, pp. 23–36, Jan. 2017, doi: 10.1007/s10845-014-0959-7.

Appendix

Power consumptions of robots per time unit (kJ)

Problem	Nm	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
P9	2	0.25	0.4	0.3	0.35	-	-	-	-	-	-	-	-	-	-
P9	3	0.3	0.35	0.3	0.4	0.25	0.4	-	-	-	-	-	-	-	-
P12	2	0.25	0.4	0.3	0.35	-	-	-	-	-	-	-	-	-	-
P12	3	0.25	0.4	0.3	0.35	0.3	0.4	-	-	-	-	-	-	-	-
P12	4	0.3	0.35	0.3	0.4	0.35	0.3	0.3	0.4	-	-	-	-	-	-
P12	5	0.3	0.35	0.3	0.4	0.35	0.3	0.3	0.4	0.3	0.4	-	-	-	-
P16	2	0.25	0.4	0.3	0.3	-	-	-	-	-	-	-	-	-	-
P16	3	0.3	0.4	0.4	0.3	0.3	0.35	-	-	-	-	-	-	-	-
P16	4	0.2	0.3	0.25	0.4	0.35	0.4	0.3	0.3	-	-	-	-	-	-
P16	5	0.3	0.2	0.3	0.25	0.25	0.35	0.4	0.4	0.3	0.25				
P24	2	0.5	0.8	0.8	0.9	-	-	-	-	-	-	-	-	-	-
P24	3	0.7	0.5	0.8	0.9	0.5	0.6	-	-	-	-	-	-	-	-
P24	4	0.8	0.9	0.5	0.9	0.7	0.5	0.8	0.6	-	-	-	-	-	-
P24	5	0.9	0.8	0.6	0.8	0.9	0.5	0.5	0.7	0.6	0.9	-	-	-	-
P65	4	1.4	1.7	1.5	1.7	1.3	1.5	1.4	1.6	-	-	-	-	-	-
P65	5	1.2	1.6	1.1	1.2	1.5	1.7	1.4	1.8	1.7	1.6	-	-	-	-
P65	6	1.5	1.7	1.1	1.4	1.1	1.2	1.5	1.7	1.4	1.8	1.7	1.3	-	-
P65	7	1.8	1.4	1.5	1.7	1.1	1.4	1.5	1.8	1.3	1.6	1.1	1.7	1.2	1.4
P65	8	1.5	1.8	1.1	1.7	1.3	1.6	1.2	1.6	1.8	1.3	1.7	1.3	1.6	1.3
P148	4	1.8	2.4	1.6	1.5	1.7	1.5	1.7	2.1	-	-	-	-	-	-
P148	5	2.1	2.2	1.8	2.4	1.6	1.5	2	2.3	2.4	2.5	-	-	-	-
P148	6	2.4	1.5	2.4	1.9	1.6	2	2.1	2	2.2	2.3	2.1	1.5	-	-
P148	7	2.2	2.1	2	2.4	1.9	2.4	1.7	1.5	1.7	2.1	2.2	1	1.7	2.3
P148	8	2.4	2.2	1.8	1.5	2.1	1.9	2.4	1.9	1.6	2	2.1	2	2.2	2.4
P148	9	2.2	1.6	1.7	2.4	1.7	1.7	2.4	2.1	2	2.2	2.3	1.9	2.1	1.8
P148	10	2.4	1.6	1.5	2.4	1.5	2.4	1.9	1.6	2	2.1	2	2.2	2.3	2.1
P148	11	1.6	2	2.1	2	2.2	1.7	2.1	2.1	2	2.2	2.3	1.9	2.1	1.9
P148	12	2.3	2.2	1.6	1.7	2.4	2.3	1.6	2	2.2	1.7	2.1	1.7	1.6	1.7
P205	4	1.8	2	2.9	1.8	2.4	2.1	2.4	2.2	-	-	-	-	-	-
P205	5	2	2.8	2	2.7	2.3	2.5	2.5	2.6	2.1	1.8	-	-	-	-
P205	6	2.1	2.9	2.5	2.8	1.9	1.9	2.4	2.3	2.5	2.2	2	2.5	-	-
P205	7	2.4	2	2.3	2.7	2.3	2.6	2.4	2.5	2.4	2.1	2.3	1.9	2	2.3
P205	8	2.2	2.5	2.1	2.7	1.9	2	2.3	2.9	2.2	2.4	2.2	2.7	1.8	2.9
P205	9	2.8	2.2	2.4	2.3	2.4	2.6	2.4	1.9	2	2.2	2.3	1.9	2.6	3
P205	10	2	2.9	1.8	2.4	2.3	2.9	2.2	2.1	2.2	2.9	2.5	2.8	1.9	1.9
P205	11	1.8	1.8	1.9	2.4	2.9	2.2	2.4	2.2	2.8	2	2.7	2.9	2.7	2.2
P205	12	2.2	2.8	2.7	2.5	2.3	1.9	2	2.3	2.9	2.3	2.6	2.4	2.5	2.4
P205	13	2.7	2.3	2.6	2.4	2.5	1.8	1.9	2.4	2.1	2.2	2	2.1	2.9	2.2
P205	14	2.3	1.9	2.3	2.1	2.2	2	2.1	2.9	2.2	2.8	2.7	2.5	2.1	2.8

Problem	Nm	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27	R28
P65	8	1.1	1.5	-	-	-	-	-	-	-	-	-	-	-	-
P148	8	1.7	1.5	-	-	-	-	-	-	-	-	-	-	-	-
P148	9	1.8	1.5	2.1	1.9	-	-	-	-	-	-	-	-	-	-
P148	10	1.5	1.9	2.1	1.8	2.1	1.7	-	-	-	-	-	-	-	-
P148	11	2.4	1.9	1.6	2.2	2.3	2.1	1.5	1.6	-	-	-	-	-	-
P148	12	1.7	2.4	2.2	1.8	1.5	2.1	1.9	1.8	2.4	1.7	-	-	-	-
P205	8	2.2	1.9	-	-	-	-	-	-	-	-	-	-	-	-
P205	9	2.7	2.5	2.5	2.5	-	-	-	-	-	-				
P205	10	1.9	2	2.3	2.9	2.2	2.4	-	-	-	-	-	-	-	-
P205	11	2.8	2.1	2.7	2.6	2.7	2.9	2.7	2.2	-	-	-	-	-	-
P205	12	1.8	1.9	2.4	2.9	2.2	2.8	2.7	2.5	2	2.9	-	-	-	-
P205	13	2.8	2.7	2.5	2.1	2.5	2.6	2.1	1.8	2.1	2.9	2.5	2.5	-	-
P205	14	3	2.1	2.4	2.7	2.9	2.2	2.1	2.2	2.8	2.1	2.7	2.6	2.7	2.9