Official URL: http://dx.doi.org/10.1080/00207543.2018.1428774

To cite this version :

Mete, Süleyman and Çil, Zeynel Abidin and Özceylan, Eren and Ağpak, Kürşad and Battaïa, Olga An optimisation support for the design of hybrid production lines including assembly and disassembly tasks. (2018) International Journal of Production Research. ISSN 0020-7543

An optimisation support for the design of hybrid production lines including assembly and disassembly tasks

Süleyman Mete[a], Zeynel Abidin Çil[b], Eren Özceylan[c], Kürşad Ağpak[d] and Olga Battaïa[e]*

[a]Department of Industrial Engineering, Munzur University, Tunceli, Turkey; [b]Department of Manufacturing Engineering, University of Batman, Batman, Turkey; [c]Department of Industrial Engineering, Gaziantep University, Gaziantep, Turkey; [d]Industrial Engineer, Gaziantep, Turkey; [e]ISAE-SUPAERO, Université de Toulouse, Toulouse, France

(*Received 8 September 2017; accepted 9 January 2018*)

The optimisation problems related to the assignment of tasks to workstations in assembly and disassembly lines have been largely discussed in the literature. They are known, respectively, as Assembly Line Balancing and Disassembly Line Balancing Problems. In this study, both types of task performed on the identical product are integrated in a common hybrid production system. Therefore, the logistic process is simplified and disassembly tasks can supply easier the assembly tasks with the required components. The considered production system has the layout of two parallel lines with common workstations. The product flow is conventional in the assembly line and reverse in the disassembly line. The paper provides a new mathematical model for designing such a hybrid system and an approximate approach based on ant colony optimisation for solving large-scale instances. The solution method is tested in a case study. The obtained results are compared with the solution provided by the design of two independent lines. The analysis of the results highlights the potential benefits of the hybrid production system.

Keywords: assembly line balancing; disassembly; mixed integer linear programming; ant colony optimisation; disassembly line balancing

1. Introduction

The increasing concern about the environment has initiated a large set of regulations on manufacturing activities and in particular, on the disposal of end-of-life products. For example, the European Union has established a series of legislative acts, known as extended producer responsibility laws, which require the manufacturers to take their responsibility for end-of-life treatment of their products and set up mandatory reuse and recycling goals for industry (Ketzenberg, Souza, and Guide 2003). Similarly, Turkish Government introduced end-of-life vehicle directive in 2011 (Demirel, Demirel, and Gökçen 2016). However, at present, only a small percentage of the value is being recovered. Many firms are still stalled by the economic instability of recovery activities, and this prevents them from recognising potential benefit. As a consequence, the disposed waste is growing as well as negative environmental impact. For instance, in Turkey, the amount of disposed waste of electrical and electronic equipment (WEEE) was about 565k tons in 2011 and is expected to reach 894k tons in 2020 (Kilic, Cebeci, and Ayhan 2015).

Successful examples of implementation of demanufacturing and remanufacturing activities also exist. A recent study of Tolio et al. (2017) presents a number of success stories of manufacturers who integrated end-of-life treatment in their traditional activities, such as Renault, Komatsu Ltd., Ricoh Company Ltd. Mitsubishi Electric Corporation to name just a few. Arçelik, one of Turkey's most representative brands and leader in white goods, established two recycling plants in Turkey within the scope of WEEE directive. The recycling plant in Eskişehir is the first recycling plant in Turkey which meets all technical requirements to disassemble used white goods (Özceylan, Paksoy, and Bektaş 2014).

In order to help the manufacturers to enhance the economic profitability of recovery activities, an important effort in developing new techniques has to be done in the domain of production research. The challenge is to take into account the different properties of reverse production flow (Battaïa and Gupta 2015; Savaskan, Bhattacharya, and Van Wassenhove 2004). This topic is attracting more and more research interest and a growing number of publications is currently dedicated to the optimisation of disassembly and reverse logistic operations (Tolio et al. 2017). While assembly is a manufacturing process in which parts are physically connected until the completion of a final product; disassembly aims to liberate constituent parts, components, subassemblies or other groupings from an end-of-life product.

*Corresponding author. Email: olga.battaia@isae.fr

Many studies show that disassembly management is a crucial stage in recovery operations (Tolio et al. 2017). However, if the integration of conventional and reverse flows is supply chain management is frequently considered together in order to design an efficient Closed-Loop Supply Chain (Battaïa and Gupta 2015), assembly and disassembly tasks have been rarely considered within an integrated system.

To the best of our knowledge, the sole study of Ketzenberg, Souza, and Guide (2003) developed the idea of the integration of assembly and disassembly operations in a hybrid production system. They used a simulation framework to analyse the efficiency of such a hybrid line and showed the potential of such a configuration. To extend this idea, in this paper, the problem of configuration of such a hybrid line is studied as a combinatorial optimisation problem with the objective to minimise the line total cost. According to our knowledge of the literature, the proposed model is the first attempt to develop an optimisation support for the design of hybrid lines including assembly and disassembly tasks. The main contributions are as follows: (i) a unified parallel assembly/disassembly line is designed, (ii) a new mathematical model is developed for solving such a hybrid system, (iii) a novel and realistic case study is presented, (iv) a heuristic based on ACO is proposed, (v) the comparison with independent layout proves the superiority of the proposed layout.

The paper is organised as follows. The results of previous research are analysed in Section 2. A new mathematical model for the design of a hybrid line with assembly and disassembly tasks is introduced in Section 3. An approximate approach based on ant colony optimisation (ACO) is developed in Section 4. A case study is presented in Section 5. A numerical experiment is conducted and analysed in Section 6. Finally, conclusions and directions for future research are given in Section 7.

2. Previous research

The Assembly Line Balancing Problem (ALBP) is a well-known combinatorial optimisation problem that has been considered in different industrial settings (Battaïa and Dolgui 2013). The first version of this problem is known as simple Assembly Line Balancing Problem (SALBP) and was introduced by Salveson (1955). The SALBP aims in minimising the number of workstations required for the assignment of a given set of tasks under precedence and cycle time constraints. Cycle time or takt time constraints verify that the sum of processing times of the tasks assigned to the same workstation is not greater than a given value c. A precedence graph G is used to model the precedence constraints. If this graph contains an arc between task i_1 and task i_2, it means that task i_1 is a direct predecessor of i_2 and this forbids assigning task i_2 to a workstation if task i_1 has not been assigned yet to a previous workstation or to the same one.

The Disassembly Line Balancing Problem (DLBP) has been introduced in early work of Gungor and Gupta (2001) and firstly was considered with a precedence graph similar to G. However, since in disassembly the precedence constraints are to be considered for separated subassemblies obtained from the initial end-of-use product, this form of graph does not suit well for the representations of all precedence relationships. To overcome this problem, Koc, Sabuncuoglu, and Erel (2009) proposed to use a transformed AND/OR Graph (TAOG) for modelling disassembly precedence constraints. Each disassembly task in TAOG is represented by a real node B and each obtained subassembly is represented by an artificial node A. An artificial node maybe preceded or succeeded by more than one normal node. It means that for each obtained subassembly, different disassembly tasks are possible but only one has to be chosen, this choice determines the possible succeeding tasks. An example of such a graph is presented in the case study in Section 5.

The differences between assembly, disassembly and machining line balancing approaches were discussed in a recent survey by Battaïa and Dolgui (2013). This survey provides a taxonomy of the problem formulations as well as an overview of the existing solution approaches for ALBP and DLBP. The extensive evolution of DLBP can be also observed through recent studies providing critical reviews on existing algorithms (Altekin, Bayındır, and Gümüşkaya 2016; Bentaha, Dolgui, and Battaïa 2015; Bentaha et al. 2014, 2015, 2018; Kalayci, Polat, and Gupta 2016; Mete, Çil, Ağpak, et al. 2016; Mete, Çil, Özceylan, et al. 2016). McGovern and Gupta (2015) presented a generalised formulae for assembly–disassembly. They developed new metrics for sequencing on assembly and disassembly lines and tested them in a case study involving the design of two separated lines. The results illustrated the inherent differences between both processes. Although the goal of assigning tasks to workstations remains, each industrial environment brings different objective functions, constraints and parameters related to the tasks.

Despite a variety of studies on assembly and disassembly line balancing problems, to the best of our knowledge, they have not been considered yet as a part of a hybrid system with common workstations. The layout of straight parallel lines has been originally studied for assembly by Gökçen, Ağpak, and Benzer (2006). For this problem, they developed a heuristic approach and mixed integer linear programming model. Further, other optimisation approaches have been proposed to solve this problem by Benzer et al. (2007), Scholl and Boysen (2009), Kara, Gökçen, and Atasagun

(2010) and Çil et al. (2017). A comprehensive review of different layouts of parallelised assembly lines was provided in a recent study of Lusa (2008).

The line balancing problem was also studied for parallel two-sided assembly lines (Ağpak and Zolfaghari 2015; Özcan, Gökçen, and Toklu 2010). Kucukkoc and Zhang (2014a, 2014b) considered this layout for mixed-model assembly lines while handling simultaneously two optimisation problems: model sequencing and line balancing. Further, Kucukkoc and Zhang (2015a) studied the parallel two-sided ALB problem with two conflicting objectives, minimising cycle time and the number of workstations simultaneously. In addition to their previous studies, Kucukkoc and Zhang (Kucukkoc and Zhang 2015b) combined U-line and parallel straight assembly line to provide an opportunity to assign tasks to multi-line workstations and to maximise resources utilisation.

Aydemir-Karadag and Turkbey (2013) treated the case of stochastic disassembly line balancing with station paralleling and developed a genetic algorithm to solve this optimisation problem.

The present study contributes to the literature by providing the first mathematical model for designing a hybrid line with parallelised assembly and disassembly flows. This model is introduced in the next section.

3. Problem definition

The proposed hybrid line has the layout shown in Figure 1. It consists of two parallel lines: one for assembly process and another one for disassembly. The manufactured products move in conventional direction at the assembly line and the end-of-usage products are transported in the reverse direction at the disassembly line. These lines have common workstations where the operators perform assembly and disassembly tasks on two different units of product as shown in Figure 1. The use of the common workstations allows reducing the total amount of equipment and tools, but also reducing the total cost of the line, since one single operator is assigned to a common workstation. Both lines are paced and have the same takt time c.

The disassembled components can be used to supply the assembly line if they pass the quality control or can be remanufactured on the shop floor.

The optimisation problem considered in this study is related to the design of such a hybrid line. The objective is to assign all given assembly and disassembly tasks respecting the precedence and takt time constraints while minimising the line cost. This cost is related to the number of workstations (if a workstation is common for both lines, its cost is the same as a simple workstation) which should be minimised, but also to the use of different equipment/tools required for executing assembly and disassembly tasks. To reduce this part of the cost, the objective is to assign the similar

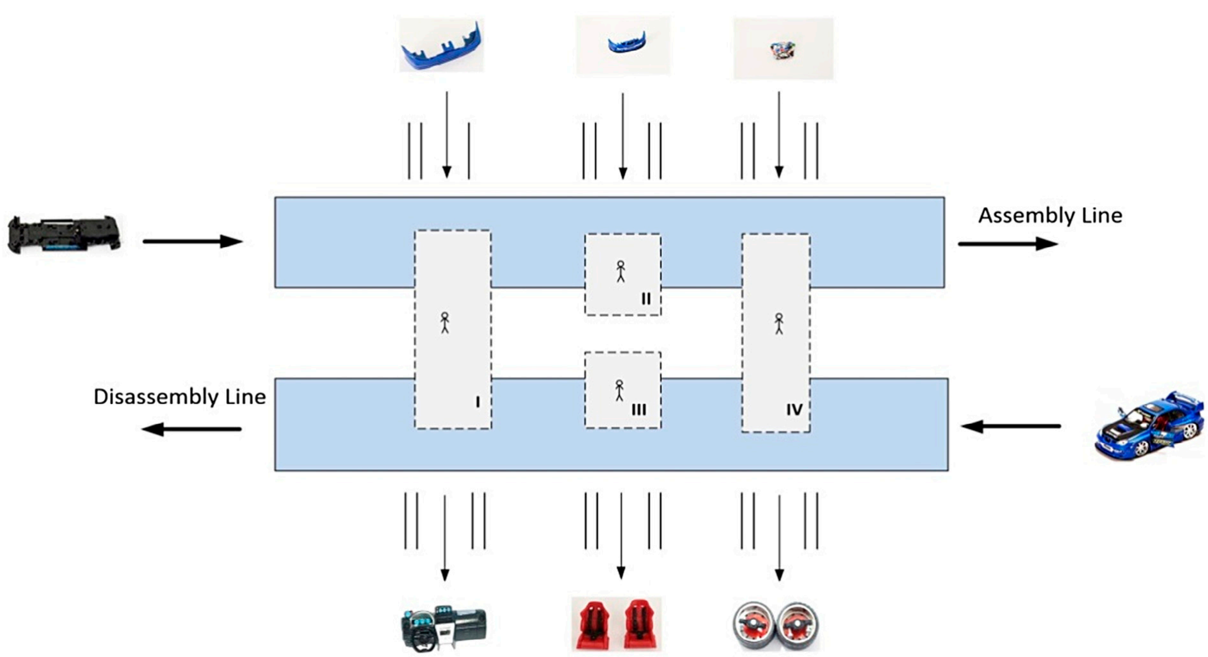

Figure 1. Schematic representation of unified assembly and disassembly line layout.

assembly and disassembly tasks (requiring the same skills or/and equipment/tools) to the same workstation. The two parts of the total cost are weighted in the objective function (1) using parameters w_1 and w_2, respectively.

The takt time constraints should be verified for all workstations and all workstations have the same takt time c. In the case of a common workstation, the sum of processing times of the tasks to be performed at both lines has to be not greater than c.

Precedence constraints for the assembly process are modelled with conventional graph G. Precedence constraints for the disassembly process are modelled with a TAOG. In this study, it is considered that each product undergoes complete disassembly.

Additionally, the following assumptions should be mentioned:

- A task cannot be split and is assigned to exactly one workstation.
- Walking times of operators between assembly and disassembly lines are ignored.
- The skills of operators and required equipment/tools are not considered explicitly, the task which are similar in their requirement to this regard are defined as 'similar' tasks, not assigning such tasks to the same workstation impose some additional cost weighted by parameter w_2.
- All workstations (even common for both lines) have the same cost.

Mathematical model

In the presented model, the following notation is used.

Indices

i	for an assembly task $i = 1 \ldots I$
j	for a disassembly task $j = 1 \ldots J$
k	for a workstation $k = 1 \ldots K$
c	cycle time (for takt time constraint)

Parameters

t_i^a	processing time of assembly task i
t_j^d	processing time of disassembly task j
A_e	artificial node of TAOG $e = 0, 1 \ldots E$
B_j	normal node in TAOG $j = 1, 2 \ldots J$
$P(A_e)$, $P(B_j)$	set of immediate predecessors of A_e, B_j, respectively
$S(A_e)$, $S(B_j)$	set of immediate successors of A_e, B_j, respectively
ST	set of pairs $g = (i, j)$ of similar assembly task i and disassembly task j
$Cost_1$	cost of one workstation
$Cost_2$	cost of not assigning of a pair of similar tasks to the same workstation

Decision variables

Z_k	1, if workstation k is opened; 0 otherwise
y_j	1, if disassembly task j is performed; 0 otherwise
d_{jk}	1, if disassembly task j is assigned to workstation k; 0 otherwise
a_{ik}	1, if assembly task i is assigned to workstation k; 0 otherwise
S_{jk}	1, if disassembly task i is chosen among the alternatives and assigned to workstation k; 0 otherwise
C_j	1, if a cost occurs because of the non-assignment of disassembly task j similar to an assembly task to the same workstation; 0 otherwise

Objective function

$$\min w_1 Cost_1 \sum_{k=1}^{K} Z_k + w_2 Cost_2 \sum_{\forall_g \in ST} C_j \tag{1}$$

Subject to

$$\sum_{j:B_j \in S(A_0)} y_j = 1 \tag{2}$$

$$\sum_{j:B_j \in S(A_e)} y_j = \sum_{j:B_j \in P(A_e)} y_j, \quad e = 1, 2 \ldots E \tag{3}$$

$$\sum_{k=1}^{K} d_{jk} = y_j, \quad j = 1, 2 \ldots J \tag{4}$$

$$\sum_{k=1}^{K} a_{ik} = 1, \quad i = 1, 2 \ldots I \tag{5}$$

$$\sum_{j:B_j \in P(A_e)} \sum_{l=1}^{k} d_{jl} \geq \sum_{j:B_j \in S(A_e)} d_{jk}, \quad e = 1, 2 \ldots E, k = 1, 2 \ldots K \tag{6}$$

$$\sum_{k=1}^{K} k a_{ik} - \sum_{k=1}^{K} k a_{fk} \leq 0, \quad \forall (i, f) \in G \tag{7}$$

$$\sum_{i=1}^{I} t_i^a a_{ik} + \sum_{j=1}^{J} t_j^d d_{jk} \leq c Z_k \quad \forall k = 1, 2 \ldots K \tag{8}$$

$$y_j + d_{jk} - 1 \leq S_{jk} \quad \forall (i, j) \in ST, \quad \forall k = 1, 2 \ldots K \tag{9}$$

$$S_{jk} + a_{ik} - 1 \leq C_j \quad \forall (i, j) \in ST, \quad \forall k = 1, 2 \ldots K \tag{10}$$

$$Z_k, y_j, a_{ik}, d_{jk}, S_{jk}, C_j \in \{0, 1\} \tag{11}$$

The objective function is defined to minimise the total number of opened workstations and the cost related to the non-assignment of similar tasks to the same workstation (1). Constraints (2) and (3) assure that exactly one of the OR-successors is selected for each disassembly task. Constraint (4) makes sure that if a task is selected it is assigned to one of the workstations. The assignment constraint (5) for assembly ensures that each task can be assigned to only one workstation. Constraint (6) sets up the precedence relations between assigned disassembly tasks (only one predecessor of A_e and only one successor can be assigned). Constraint (7) provides precedence relationships for assembly. Constraint (8) ensures that the total processing time for assembly and disassembly tasks in each workstation cannot exceed the cycle time. Constraints (9) determines if a disassembly task having a similar assembly task has been assigned in the solution and if yes, constraint (10) determines if the assembly similar task has been assigned to the same workstation, otherwise decision variable C_j will bring the corresponding cost to the objective function. Constraints (11) indicate binary nature of decision variables.

The proposed model can be used for solving small size problem instances, but since ALBP and DLBPS were proven to be NP-hard (Battaïa and Dolgui 2013), an approximate algorithm is required for tackling large-scale problem instances. Such an algorithm based on ACO is developed in the next section.

4. Ant colony optimisation algorithm

In the literature, ACO has already shown good performances for solving ALBP e.g. (Bautista and Pereira 2007; Sabuncuoglu, Erel, and Alp 2009; Vilarinho and Simaria 2006) and DLBP, e.g. (Agrawal and Tiwari 2008; Ding et al. 2010; Kalayci and Gupta 2013). For this reason, this algorithm has been selected for addressing the optimisation problem for designing hybrid production lines with assembly and disassembly tasks.

The developed algorithm consists of three steps described here below.

Step 1. The following parameters are set up: heuristic information (α, β, Q), pheromone evaporating parameters (ρ_1, ρ_2), initial pheromone level (τ_0), user-defined parameters (r_1, r_2), number of ants, number of colonies.

Step 2. A solution is constructed workstation by workstation. Because of the opposite directions of product flows in the hybrid system, the solution construction starts from the 'last' assembly workstation which is also the potentially 'first' one for the disassembly line. Therefore, the tasks are considered from the end of assembly precedence graph and from the beginning of TAOG. When a new workstation is open, a list of available tasks is constructed for assembly and disassembly lines taking into account the respective precedence constraints and the available workstation time. Then, the tasks to be assigned to the current workstation are selected from these lists according to the priority values. The priority value for assembly and disassembly tasks is computed using different procedures (Equations (12) and (13)). The priority value η_i of assembly task i can be calculated as follows:

$$\eta_i = \frac{t_i^a}{c} + \frac{P_i}{\max P} \tag{12}$$

η_i	priority value of assembly task i,
t_i^a	task time of assembly operation i,
p_i	number of all predecessors of assembly task i,
$\max P$	number of the predecessors of the task having the maximum number of the predecessors among all assembly tasks.

As a result of pre-tests, this priority rule gives poor results for disassembly tasks because of alternative precedence relationships. Hence, a new priority rule is developed for disassembly:

$$\eta_j = \frac{\left[\sum_{j\in AT}\left(t_j^d + \min_{v\in S(B_j)}\left(t_v^d\right)\right)\right] - \left[t_j^d + \min_{v\in S(B_j)}\left(t_v^d\right)\right]}{C} \tag{13}$$

where

η_j	priority value of disassembly task j,
AT	list of assignable disassembly tasks,
V	a successor of task j,
$S(B_j)$	set of all immediate successors of B_j,
t_j^d	processing time of disassembly task j.

The selection procedure uses a random number r between 0 and 1 and two user-defined parameters r_1, r_2. The selection of a task from the set of available tasks is performed in the following way:

$$j = \begin{cases} \arg\max_{j\in AT}\left\{\left(\tau_{jk}\right)^{\alpha}\left(\eta_j\right)^{\beta}\right\} & \text{if } 0 \le r < r_1 \\ \rho_{ik} = \dfrac{\left(\tau_{jk}\right)^{\alpha}\left(\eta_j\right)^{\beta}}{\sum_{p\in AT}\left(\tau_{pk}\right)^{\alpha}\left(\eta_p\right)^{\beta}} & \text{if } r_1 \le r < r_1 + r_2 \\ \text{Random selection } j \in AT & \text{if } r_1 + r_2 \le r \le 1 \end{cases} \tag{14}$$

The selection of a task from the set of available tasks is performed by one of three strategies. Firstly, one selects the best task according to the values of $(\tau_{jk})^{\alpha}(\eta_j)^{\beta}$. Secondly, a task is selected according to probability of ρ_{ik}. Lastly, from the set of available task list (AT), an ant selects a task randomly. When no task can be assigned to the current workstation, a new one is open, and the algorithm continues until all tasks are assigned.

Step 3. The pheromone release strategy is based on the one used by Dorigo and Blum (2005). There are two basic rules for updating the pheromone called local and global updating rules. Following the allocation of tasks to workstation k, the pheromone level is updated using Equation (15):

$$\tau_{ik} = (1 - \rho_1)\tau_{ik} + \rho_1\tau_0 \tag{15}$$

where ρ_1 is the local pheromone evaporating parameter $(0 \le \rho_1 \le 1)$; τ_0 is the initial pheromone level. The global update is performed after a colony is completed a tour according to Equation (16) as follows:

$$\tau_{ik} = (1 - \rho_2)\tau_{ik} + \rho_2\Delta\tau_{ik} \tag{16}$$

where $0 \le \rho_2 \le 1$ is the pheromone evaporation parameter;

$$\Delta\tau_{ik} = \begin{cases} \frac{Q}{bs} & \text{if } (i,k) \in \text{best} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where Q is a constant; bs is the number of workstation in the best known solution. The overall procedure of ACO is implemented as follows:

```
Initialize Parameters
  for each colony
      Set initial inputs and parameters
    for each ant
      while available task list (AT) ≠ ∅
      Determine AT list
        ifAT ∈ ∅
        Current workstation = Current workstation + 1
          Remaining time = Cycle time
            AT = Available task list
        end if
      Select a task j according to the following task assignment strategy:
```

$$j = \begin{cases} argmax_{j \in AT}\left\{ \left(\tau_{jk}\right)^{\alpha} \left(\eta_j\right)^{\beta} \right\} & if\ 0 \leq r < r_1 \\ \rho_{ik} = \dfrac{\left(\tau_{jk}\right)^{\alpha}\left(\eta_j\right)^{\beta}}{\sum_{p \in AT} \left(\tau_{pk}\right)^{\alpha}\left(\eta_p\right)^{\beta}} & if\ r_1 \leq r < r_1 + r_2 \\ Random\ selection\ j \in AT & if\ r_1 + r_2 \leq r \leq 1 \end{cases}$$

```
        Update the local pheromone
```

$$\tau l_{ik} = (1 - \rho_1)\tau l_{ik} + \rho_1 \tau_0$$

```
        Update the Precedence matrix,
          Remaining time = Remaning time − Opreation time of task j
      end while
    end for
    update the global pheromone
```

$$\tau_{ik} = (1 - \rho_2)\tau_{ik} + \rho_2 \Delta\tau_{ik}$$

```
    Initial local pheromone (τl_ik) = global pheromone (τ_ik)
      if local best solution is better than best solution (found up to now)
        update best solution
      end if
  end for
```

5. Case study

This section presents a case study of a toy car for which a hybrid line with assembly and disassembly tasks is designed. The toy car has 1/24 scale of the original car. The product consists of 12 components shown in Figure 2.

Assembly precedence diagram of the product is presented in Figure 3 where numbers with circles denote the assembly tasks detailed in Table 1. Processing times are also given in Table 1.

Disassembly AND/OR graph for this product is presented in Figure 4. There are 43 different possible subassemblies and 97 possible disassembly tasks. For simplicity, subassemblies with only one component are not shown in the graph. Each node corresponding to a subassembly is represented by an artificial node in the TAOG and each disassembly task is represented by a normal node in TAOG. To differentiate between the AND-type and OR-type relations, a small curve is put as an indicator of OR-type relations in Figure 4. The 97 disassembly tasks are described in Table 2. As can be seen from Table 2, some disassembly tasks (except tasks 48, 74 and 97) result in the same subassemblies or parts, but they do not necessarily have the same processing time as indicated in Table 3.

As mentioned in the problem definition section, some of the assembly and disassembly tasks are similar. For instance, while assembly task (1) is assembling of the back wheels (No: 11) with the undercarriage (No: 12);

Table 1. Assembly data of toy car instance.

Task	Description	Time (s)
1	Assemble the back wheels (No:11) with the undercarriage (No:12)	60
2	Assemble the front wheels (No:8) with the subassembly of the back wheels (No:11) and undercarriage (No:12)	70
3	Assemble the seats (No:10) with the gear and base (No:9)	46
4	Assemble the subassembly of the seats (No:10) and gear and base (No:9) with the subassembly of the front wheels (No:8), back wheels (No:11) and undercarriage (No:12)	15
5	Assemble the doors (No:4) with the main body (No:7)	25
6	Assemble the front console (No:5) with the subassembly the doors (No:4) and main body (No:7)	13
7	Assemble the front bumper (No:1) with the subassembly of the doors (No:4), front console (No:5) and main body (No:7)	13
8	Assemble the back bumper (No:6) with the subassembly of the front bumper (No:1), doors (No:4), front console (No:5) and main body (No:7)	20
9	Assemble the subassembly seats (No:10), gear and base (No:9), front wheels (No:8), back wheels (No:11) and undercarriage (No:12) with the subassembly back bumper (No:6), front bumper (No:1), doors (No:4), front console (No:5), main body (No:7)	40
10	Assemble the engine (No:2) with the subassembly seats (No:10), gear and base (No:9), front wheels (No:8), back wheels (No:11), undercarriage (No:12), back bumper (No:6), front bumper (No:1), doors (No:4), front console (No:5), main body (No:7)	1
11	Assemble the hood (No:3) with the subassembly of engine (No:2), seats (No:10), gear and base (No:9), front wheels (No:8), back wheels (No:11), undercarriage (No:12), back bumper (No:6), front bumper (No:1), doors (No:4), front console (No:5), main body (No:7)	2

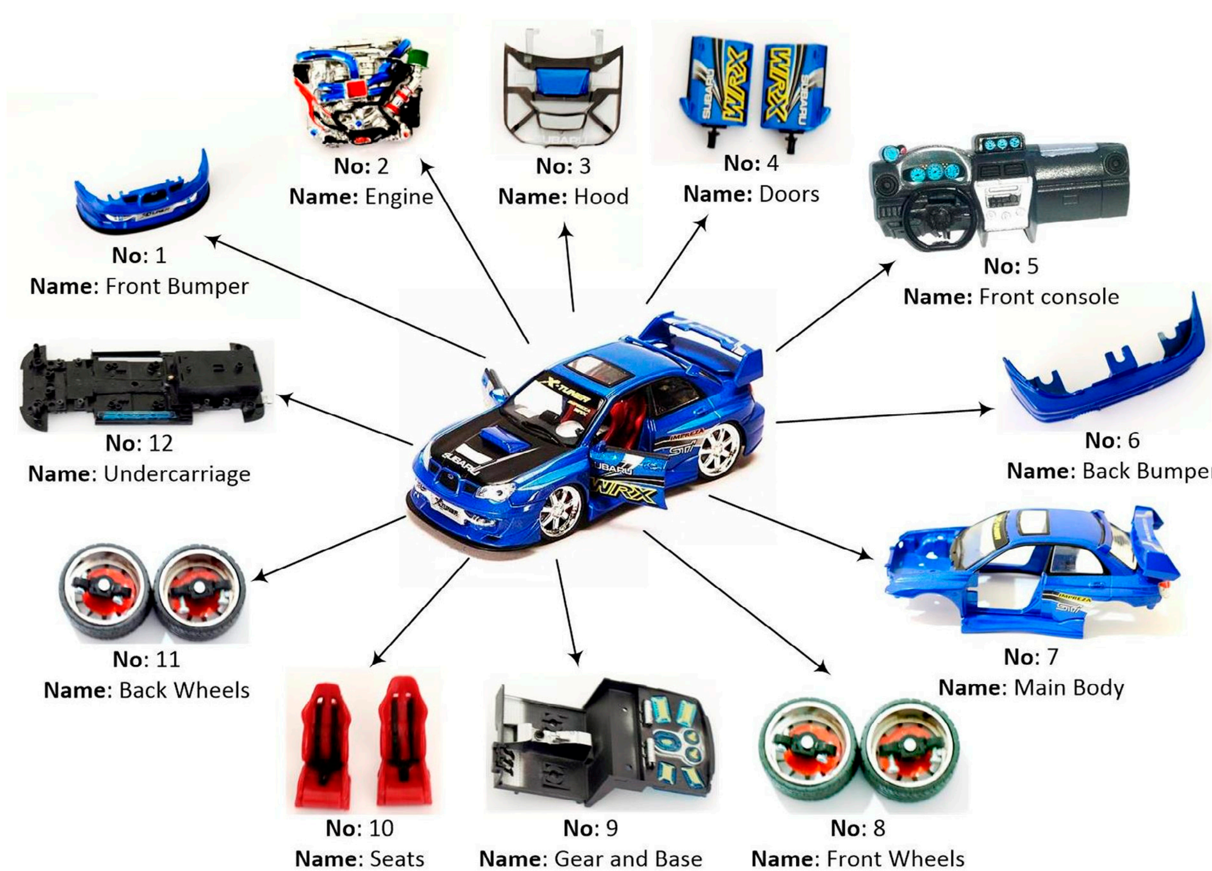

Figure 2. Considered toy car and its components.

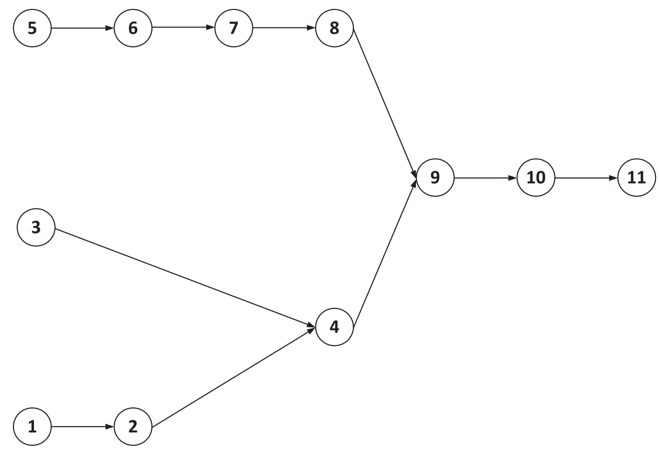

Figure 3. Assembly precedence diagram of the product.

disassembling of the back wheels (No: 11) from the undercarriage (No: 12) is performed by disassembly tasks (89) and (96). In this case, assembly task (1) and disassembly tasks (89) and (96) are considered 'similar'. Such similar assembly and disassembly tasks are depicted in Table 4.

The different cycle times were used in the numerical example (75, 80, 96 and 120 s). The maximum number of workstations was limited by 10. The model (1)–(11) with aforementioned parameters was first solved with a commercial solver GAMS-CPLEX. All computational experiments were conducted on a PC with an Intel I3 3.10 GHz processor with 4 GB of RAM. Detailed results which represent opened workstations, assigned tasks, overall line efficiencies and CPU times are illustrated in Figure 5.

The required computation times to solve the model to optimality using GAMS-CPLEX were 19, 47, 10 and 9 s, respectively. As was expected, increasing the cycle time decreased the number of workstations and the solution time. Overall line efficiency of the solutions was calculated as 1 − (total idle time ÷ (number of workstations × cycle time)). The overall line efficiency varied between 79 and 93%.

In Figure 5, while circles represent assembly tasks, disassembly tasks are represented with squares. Grey coloured circles and squares indicate the similar tasks which are assigned to different workstations. On the other hand, similar tasks which are assigned to the same workstations are coloured with blue. Tasks without a colour mean that they do not have a similar task which is assigned to a workstation. Apart two cases, all similar assembly and disassembly tasks were assigned to the same workstations.

6. Numerical experiment

The objective of this section is two-fold. The first one is to compare the solutions obtained while designing two independent lines: one assembly and one disassembly with the solutions provided by the design method developed in this study where assembly and disassembly lines are combined in a hybrid line with common workstations. The second goal is to compare the performances of the commercial solver GAMS-CPLEX finding an exact solution for the corresponding optimisation problem and the developed approximate ACO method implemented in MATLAB.

The numerical experiment was conducted on eight main problem instances. The first of them was created for this study; other seven were taken from the literature. All of them were considered with different cycle times resulting in 29 different problem test instances. The number of assembly tasks in these eight problems varied between 7 and 45; the number of disassembly tasks between 70 and 640, i.e. the total number of tasks was between 77 and 685. All data about the problem are uploaded at the URL link and can be downloaded from: https://drive.google.com/drive/folders/1lQD9geDB3GK2rih3xtCM1dCi2UoL2yng?usp=sharing.

The solution time was limited to 3600 s for GAMS-CPLEX. The input parameters of the heuristic algorithm were as follows: $\alpha = 1$, $\beta = 2$, $\rho_1 = 0.9$, $\rho_2 = 0.9$, $Q = 1$, $\tau_0 = 0.5$, the number of iterations was limited to 500 and the number of ants (population size) was limited to 100. Each test problem was solved 10 times using the heuristic algorithm and the best values are reported. Since the similarity between assembly and disassembly tasks has no sense for the case of independent lines, this property was ignored in the numerical experiment.

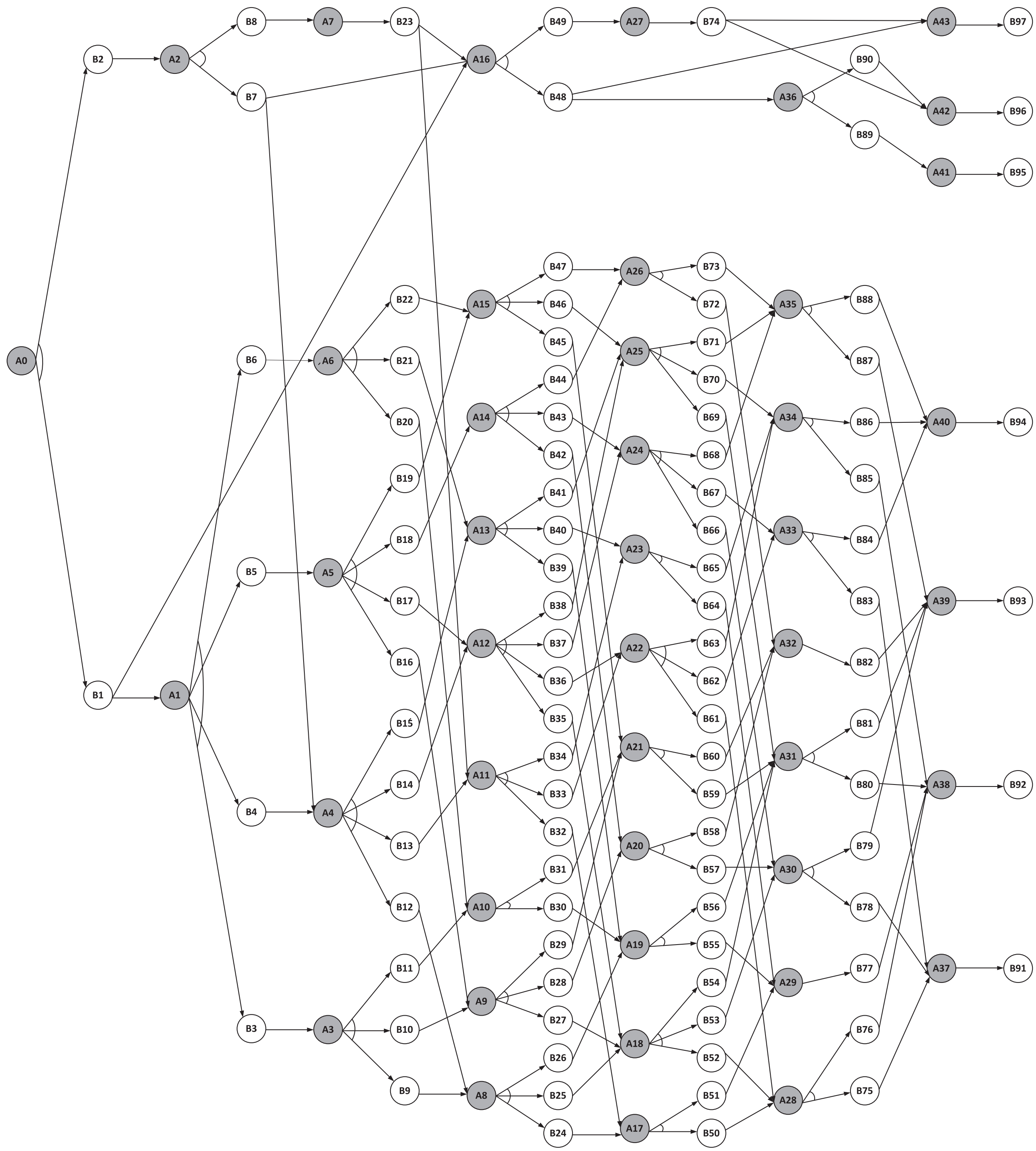

Figure 4. The transformed AND/OR graph of the product.

Table 5 shows the obtained results. The improvement rate (Imp %) is computed using the following formula: 100 × (No. of workstations for two independent lines − No. of workstations in the hybrid line)/(No. of workstations for two independent lines). The improvement rate varies from 0 to 25% for all problems and average improvement rate is 7.45%. In 21 of the 29 test problems, the number of workstations in the hybrid line was decreased in comparison with

Table 2. Description of disassembly tasks.

Disassembly task	Description
1, 7, 23	Disassemble the subassembly consisting of front wheels (No:8), gear and base (No:9), seats (No:10), back wheels (No:11) and the undercarriage (No:12) from the product
2, 4, 9, 17, 21, 27, 30, 43, 46, 57, 59, 73, 82	Disassemble the hood (No:3) from the main body (No:7)
3, 12, 16, 20, 32, 35, 39, 42, 45, 61, 64, 66, 69, 72, 83, 85, 87, 94	Disassemble the front bumper (No:1) from the main body (No:7)
5, 10, 14, 22, 25, 31, 33, 41, 50, 56, 65, 77	Disassemble the front console (No:5) from the main body (No:7)
8, 13, 24, 36, 40, 52, 55, 67, 70, 78, 80, 88, 93	Disassemble the engine (No:2) from the main body (No:7)
18, 28, 37, 47, 53, 60, 62, 71, 75, 81, 86, 92	Disassemble the doors (No:4) from the main body (No:7)
6, 11, 15, 19, 26, 29, 34, 38, 44, 51, 54, 58, 63, 68, 76, 79, 84, 91	Disassemble the back bumper (No:6) from the main body (No:7)
49, 90, 95	Disassemble the front wheels (No:8) from the undercarriage (No:12)
97	Disassemble the seats (No:10) from the gear and base (No:9)
89, 96	Disassemble the back wheels (No:11) from the undercarriage (No:12)
48	Disassemble two subassemblies: (*i*) consisting of the gear and base (No:9) and seats (No:10) and (*ii*) consisting of the front wheels (No:8), back wheels (No:11) and the undercarriage (No:12)
74	Disassemble two subassemblies: (*i*) consisting of the gear and base (No:9) and seats (No:10) and (*ii*) consisting of the back wheels (No:11) and the undercarriage (No:12)

Table 3. Task times of disassembly operations of toy car instance.

Task	Time (s)	Task	Time (s)	Task	Time (s)	Task	Time (s)	Task	Time (s)	Task	Time (s)
1	37	18	19	35	5	52	2	69	5	86	19
2	3	19	6	36	2	53	19	70	2	87	5
3	5	20	5	37	19	54	6	71	19	88	2
4	3	21	3	38	6	55	2	72	5	89	29
5	10	22	10	39	5	56	10	73	3	90	34
6	6	23	37	40	2	57	3	74	11	91	6
7	37	24	2	41	10	58	6	75	19	92	19
8	2	25	10	42	5	59	3	76	6	93	2
9	3	26	6	43	3	60	19	77	10	94	5
10	10	27	3	44	6	61	5	78	2	95	34
11	6	28	19	45	5	62	19	79	6	96	29
12	5	29	6	46	3	63	6	80	2	97	14
13	2	30	3	47	19	64	5	81	19		
14	10	31	10	48	11	65	10	82	3		
15	6	32	5	49	34	66	5	83	5		
16	5	33	10	50	10	67	2	84	6		
17	3	34	6	51	6	68	6	85	5		

two independent lines. For these cases, the average improvement rate is 10.28%. In the remaining eight test problems, the number of workstations is identical for both solutions.

The ACO approach could provide the optimal results (shown in bold) for the first 17 problems for which the optimal solution is known. For remaining 12 problems, the commercial solver didn't achieve a feasible solution. Not surprisingly, ACO method requires less computation time than the CPLEX, this time is within 15 s of CPU time on average.

Table 4. Similar assembly and disassembly tasks; obtained parts after disassembly task.

Assembly task	Disassembly task	Obtained part(s) after disassembly task
1	89, 96	No:11 and No:12
2	49, 90, 95	No:8
3	97	No:9 and No:10
4	48	NA
5	18, 28, 37, 47, 53, 60, 62, 71, 75, 81, 86, 92	No:4 and No:7
6	5, 10, 14, 22, 25, 31, 33, 41, 50, 56, 65, 77	No:5
7	3, 12, 16, 20, 32, 35, 39, 42, 45, 61, 64, 66, 69, 72, 83, 85, 87, 94	No:1
8	6, 11, 15, 19, 26, 29, 34, 38, 44, 51, 54, 58, 63, 68, 76, 79, 84, 91	No:6
9	NA	NA
10	8, 13, 24, 36, 40, 52, 55, 67, 70, 78, 80, 88, 93	No:2
11	2, 4, 9, 17, 21, 27, 30, 43, 46, 57, 59, 73, 82	No:3

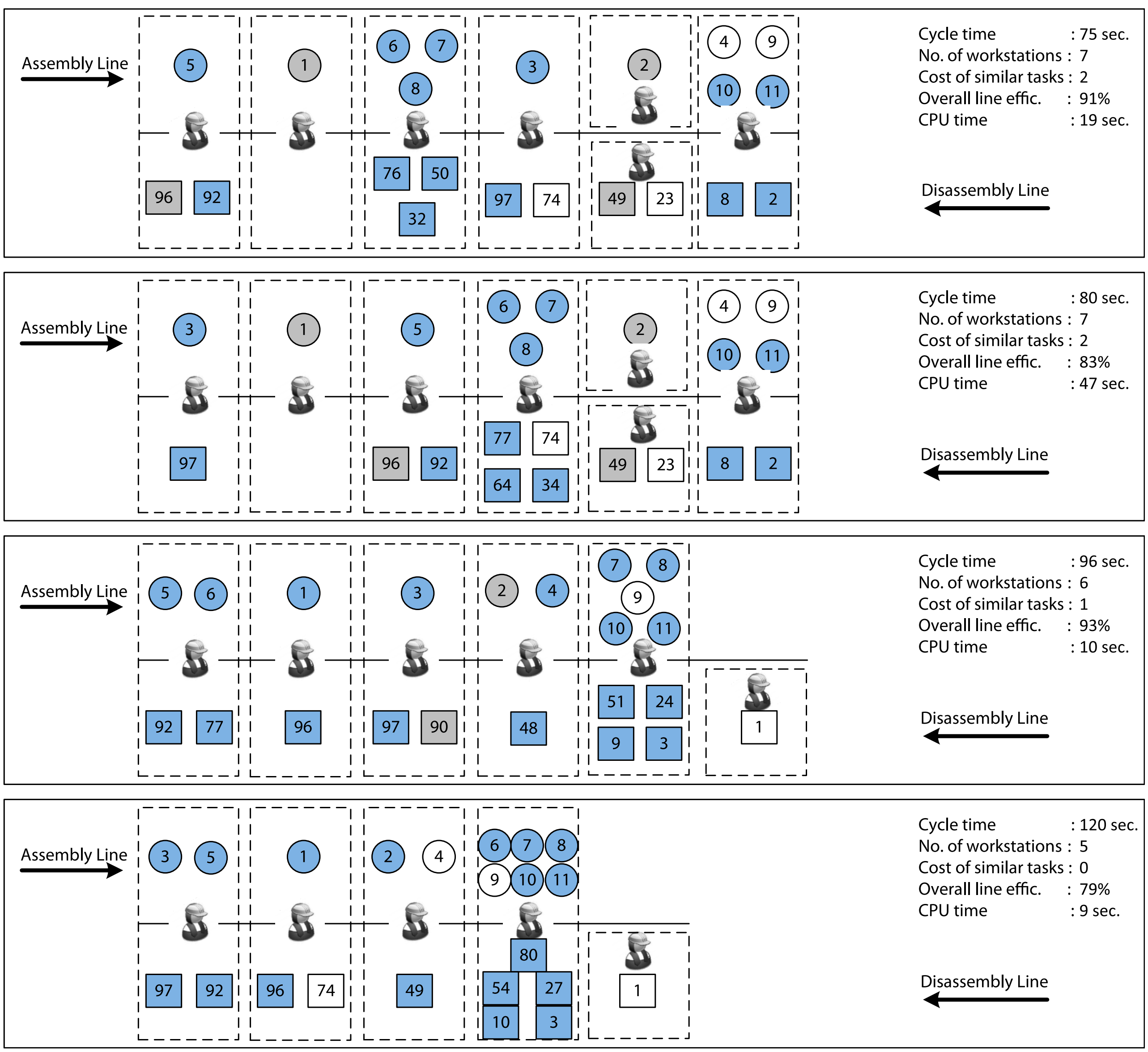

Figure 5. The optimal line layouts for each cycle time.

Table 5. Experimental results for independent and parallel line balances.

Test problems											
Assembly line (no. of task)	Disassembly line $P(a–t–N)$	A	B	Total no. of task	Cycle time	No. of workstations for independent balance	Results of CPLEX for the unified design	CPU (s) of CPLEX	Results of heuristic for the unified design	CPU (s) of heuristic	Imp (%)
Toy car (11)	–	43	97	108	75	5 + 3 = 8	**7**	<1	**7**	<1	12.50
					80	5 + 3 = 8	**6**	<1	**6**	<1	25.00
					96	4 + 2 = 6	**5**	<1	**5**	<1	16.60
					120	3 + 2 = 5	**4**	<1	**4**	<1	20.00
Mertens (7)	(5–3–7)	26	70	77	6	6 + 3 = 9	**8**	<1	**8**	<1	11.00
					7	5 + 3 = 8	**7**	<1	**7**	<1	12.50
					8	5 + 2 = 7	**6**	<1	**6**	<1	14.20
					10	3 + 2 = 5	**5**	<1	**5**	<1	0.00
Bowman (8)	(5–3–8)	31	85	93	20	5 + 3 = 8	**7**	<1	**7**	<1	12.50
Mitchell (21)	(5–3–21)	96	280	301	14	8 + 5 = 13	**13**	333	**13**	12	0.00
					15	8 + 5 = 13	**12**	20	**12**	12	7.60
					26	5 + 3 = 8	**7**	15	**7**	10	12.50
					35	3 + 2 = 5	**5**	4	**5**	9	0.00
Roszieg (25)	(5–3–25)	116	340	365	14	10 + 6 = 16	**15**	271	**15**	16	6.25
					16	8 + 6 = 14	**13**	164	**13**	14	7.10
					18	8 + 5 = 13	**12**	48	**12**	14	7.60
					21	6 + 4 = 10	**10**	228	**10**	13	0.00
Sawyer (30)	(5–3–30)	141	415	445	25	14 + 9 = 23	22*	+3600	22	21	4.30
					27	13 + 8 = 22	**	+3600	21	20	4.50
					33	11 + 7 = 18	17*	+3600	17	19	5.50
					36	10 + 6 = 16	**	+3600	15	18	6.20
Gunther (35)	(5–3–35)	166	490	525	41	14 + 8 = 22	**	+3600	21	24	4.50
					61	9 + 5 = 14	**	+3600	14	20	0.00
					69	8 + 5 = 13	**	+3600	12	19	7.60
					81	7 + 4 = 11	**	+3600	10	18	9.00
Kilbridge (45)	(5–3–45)	216	640	685	56	10 + 10 = 20	**	+3600	20	29	0.00
					62	9 + 9 = 18	**	+3600	18	27	0.00
					69	8 + 8 = 16	**	+3600	16	26	0.00
					110	6 + 5 = 11	**	+3600	10	23	9.00

*Integer solution.
**No solution.

7. Conclusion

While forward and reverse flows in supply chains are often jointly considered within a closed-loop supply chain, the idea of the integration of assembly and disassembly tasks in a hybrid production system with common workstations is relatively new. This study offers the first mathematical model for balancing such a hybrid line while considering the similarity between the assembly and disassembly tasks. Source of these similarities could be the tool requirements, operation similarity or close operator skills. In a general case, the assignment of such tasks to the same workstation helps to reduce the total line cost. In addition to the mathematical model, a new ACO-based approach is developed in this study in order to address the large-scale problem instances.

A case study and a numerical experiment were used to validate the proposed solution methods. The results were also compared with the line configurations obtained while designing two independent lines. The use of a hybrid line with common workstations between assembly and disassembly lines may help to reduce the total number of workstations. Obviously, this layout is only realisable when both lines can work at same takt time, which is currently relatively challenging in the industry. However, the development of recovery operations and the progress in production research will help to optimise the reuse of the existing components and may follow to the development of such configurations especially for complex high-value-added products.

Our study can be extended in several directions. For example, it is important to consider the integration of the disassembled components/parts in feeding of the assembly line. Further, since the disassembly of end-of-life products is characterised by a high level of uncertainty and high variety of treated products, these aspects should be also considered in the future design procedures for hybrid lines.

Acknowledgements

The authors express sincere appreciation to the editor and anonymous reviewers for their efforts to improve the quality of this paper. Third author was supported by the BAGEP Award of the Science Academy in Turkey.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Zeynel Abidin Çil http://orcid.org/0000-0002-7270-9321
Olga Battaïa http://orcid.org/0000-0002-5367-7846

References

Ağpak, K., and S. Zolfaghari. 2015. "Mathematical Models for Parallel Two-Sided Assembly Line Balancing Problems and Extensions." *International Journal of Production Research* 53 (4): 1242–1254.

Agrawal, S., and M. K. Tiwari. 2008. "A Collaborative Ant Colony Algorithm to Stochastic Mixed-Model U-Shaped Disassembly Line Balancing and Sequencing Problem." *International Journal of Production Research* 46 (6): 1405–1429.

Altekin, F. T., Z. P. Bayındır, and V. Gümüşkaya. 2016. "Remedial Actions for Disassembly Lines with Stochastic Task times." *Computers & Industrial Engineering* 99: 78–96.

Aydemir-Karadag, A., and O. Turkbey. 2013. "Multi-Objective Optimization of Stochastic Disassembly Line Balancing with Station Paralleling." *Computers & Industrial Engineering* 65 (3): 413–425.

Battaïa, O., and A. Dolgui. 2013. "A Taxonomy of Line Balancing Problems and Their Solutionapproaches." *International Journal of Production Economics* 142: 259–277.

Battaïa, O., and S. M. Gupta. 2015. "Reverse Supply Chains: A Source of Opportunities and Challenges." *Journal of Manufacturing Systems* 37: 587–588.

Bautista, J., and J. Pereira. 2007. "Ant Algorithms for a Time and Space Constrained Assembly Line Balancing Problem." *European Journal of Operational Research* 177 (3): 2016–2032.

Bentaha, M. L., O. Battaïa, A. Dolgui, and S. J. Hu. 2014. "Dealing with Uncertainty in Disassembly Line Design." *CIRP Annals – Manufacturing Technology* 63 (1): 21–24.

Bentaha, M. L., O. Battaïa, A. Dolgui, and S. J. Hu. 2015. "Second Order Conic Approximation for Disassembly Line Design with Joint Probabilistic Constraints." *European Journal of Operational Research* 247 (3): 957–967.

Bentaha, M. L., A. Dolgui, and O. Battaïa. 2015. "A Bibliographic Review of Production Line Design and Balancing Under Uncertainty." In *Proceedings of the 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'15)*, Ottawa, Canada, 70–75. IFAC-PapersOnline.net. Elsevier Science. ISSN 1474-6670.

Bentaha, M. L., A. Dolgui, O. Battaïa, R. J. Riggs, and J. Hu. 2018. "Profit-oriented Partial Disassembly Line Design: Dealing with Hazardous Parts and Task Processing Times Uncertainty." *International Journal of Production Research*, 1–23. doi:10.1080/00207543.2017.1418987.

Benzer, R., H. Gökçen, T. Çetinyokus, and H. Çerçioglu. 2007. "A Network Model for Parallel Line Balancing Problem." *Mathematical Problems in Engineering* 2007: 101–106.

Çil, Z. A., S. Mete, E. Özceylan, and K. Agpak. 2017. "A Beam Search Approach for Solving Type II Robotic Parallel Assembly Line Balancing Problem." *Applied Soft Computing* 61: 129–138.

Demirel, E., N. Demirel, and H. Gökçen. 2016. "A Mixed Integer Linear Programming Model to Optimize Reverse Logistics Activities of End-of-Life Vehicles in Turkey." *Journal of Cleaner Production* 112 Part 3: 2101–2113.

Ding, L.-P., Y.-X. Feng, J.-R. Tan, and Y.-C. Gao. 2010. "A New Multi-Objective Ant Colony Algorithm for Solving the Disassembly Line Balancing Problem." *The International Journal of Advanced Manufacturing Technology* 48 (5–8): 761–771.

Dorigo, M., and C. Blum. 2005. "Ant Colony Optimization Theory: A Survey." *Theoretical Computer Science* 344 (2–3): 243–278.

Gökçen, H., K. Ağpak, and R. Benzer. 2006. "Balancing of Parallel Assembly Lines." *International Journal of Production Economics* 103 (2): 600–609.

Gungor, A., and S. M. Gupta. 2001. "A Solution Approach to the Disassembly Line Balancing Problem in the Presence of Task Failures." *International Journal of Production Research* 39 (7): 1427–1467.

Kalayci, C. B., and S. M. Gupta. 2013. "Ant Colony Optimization for Sequence-Dependent Disassembly Line Balancing Problem." *Journal of Manufacturing Technology Management* 24 (3): 413–427.

Kalayci, C. B., O. Polat, and S. M. Gupta. 2016. "A Hybrid Genetic Algorithm for Sequence-Dependent Disassembly Line Balancing Problem." *Annals of Operations Research* 242 (2): 321–354.

Kara, Y., H. Gökçen, and Y. Atasagun. 2010. "Balancing Parallel Assembly Lines with Precise and Fuzzy Goals." *International Journal of Production Research* 48 (6): 1685–1703.

Ketzenberg, M. E., G. C. Souza, and V. D. R. Guide. 2003. "Mixed Assembly and Disassembly Operations for Remanufacturing." *Production and Operations Management* 12 (3): 320–335.

Kilic, H. S., U. Cebeci, and M. B. Ayhan. 2015. "Reverse Logistics System Design for the Waste of Electrical and Electronic Equipment (WEEE) in Turkey." *Resources, Conservation and Recycling* 95: 120–132.

Koc, A., I. Sabuncuoglu, and E. Erel. 2009. "Two Exact Formulations for Disassembly Line Balancing Problems with Task Precedence Diagram Construction Using an aND/oR Graph." *IIE Transactions* 41 (10): 866–881.

Kucukkoc, I., and D. Z. Zhang. 2014a. "Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines." *International Journal of Production Research* 52 (12): 3665–3687.

Kucukkoc, I., and D. Z. Zhang. 2014b. "Mathematical Model and Agent Based Solution Approach for the Simultaneous Balancing and Sequencing of Mixed-Model Parallel Two-Sided Assembly Lines." *International Journal of Production Economics* 158: 314–333.

Kucukkoc, I., and D. Z. Zhang. 2015a. "A Mathematical Model and Genetic Algorithm-Based Approach for Parallel Two-Sided Assembly Line Balancing Problem." *Production Planning & Control* 26 (11): 874–894.

Kucukkoc, I., and D. Z. Zhang. 2015b. "Balancing of Parallel U-Shaped Assembly Lines." *Computers & Operations Research* 64: 233–244.

Lusa, A. 2008. "A Survey of the Literature on the Multiple or Parallel Assembly Line Balancing Problem." *European Journal of Industrial Engineering* 2 (1): 50–72.

McGovern, S. M., and S. M. Gupta. 2015. "Unified Assembly-and Disassembly-Line Model Formulae." *Journal of Manufacturing Technology Management* 26 (2): 195–212.

Mete, S., Z. A. Çil, K. Ağpak, E. Özceylan, and A. Dolgui. 2016. "A Solution Approach Based on Beam Search Algorithm for Disassembly Line Balancing Problem." *Journal of Manufacturing Systems* 41: 188–200.

Mete, S., Z. A. Çil, E. Özceylan, and K. Ağpak. 2016. "Resource Constrained Disassembly Line Balancing Problem." *IFAC-PapersOnLine* 49 (12): 921–925.

Özcan, U., H. Gökçen, and B. Toklu. 2010. "Balancing Parallel Two-Sided Assembly Lines." *International Journal of Production Research* 48 (16): 4767–4784.

Özceylan, E., T. Paksoy, and T. Bektaş. 2014. "Modeling and Optimizing the Integrated Problem of Closed-Loop Supply Chain Network Design and Disassembly Line Balancing." *Transportation Research Part E: Logistics and Transportation Review* 61: 142–164.

Sabuncuoglu, I., E. Erel, and A. Alp. 2009. "Ant Colony Optimization for the Single Model U-Type Assembly Line Balancing Problem." *International Journal of Production Economics* 120 (2): 287–300.

Salveson, M. 1955. "The Assembly Line Balancing Problem." *Journal of Industrial Engineering* 6 (3): 18–25.

Savaskan, R. C., S. Bhattacharya, and L. N. Van Wassenhove. 2004. "Closed-Loop Supply Chain Models with Product Remanufacturing." *Management Science* 50 (2): 239–252.

Scholl, A., and N. Boysen. 2009. "Designing Parallel Assembly Lines with Split Workplaces: Model and Optimization Procedure." *International Journal of Production Economics* 119 (1): 90–100.

Tolio, T., A. Bernard, M. Colledani, S. Kara, G. Seliger, J. Duflou, O. Battaia, and S. Takata. 2017. "Design, Management and Control of Demanufacturing and Remanufacturing Systems." *CIRP Annals – Manufacturing Technology* 66 (2): 585–609.

Vilarinho, P. M., and A. S. Simaria. 2006. "ANTBAL: An Ant Colony Optimization Algorithm for Balancing Mixed-Model Assembly Lines with Parallel Workstations." *International Journal of Production Research* 44 (2): 291–303.