

Original research article

## Combining Line Balancing Methods and Discrete Event Simulation: A Case Study from a Metalworking Company

N. Mirzaei <sup>a\*</sup>, M. G. Nejad <sup>b</sup>, N. O. Fernandes <sup>c</sup>

<sup>a</sup> Istanbul Aydin University, Department of Industrial Engineering, Istanbul, Turkey;

<sup>b</sup> Girne American University, Industrial Engineering Department, Kyrenia, TRNC, Turkey;

<sup>c</sup> Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal

### ABSTRACT

In this study, we consider a real-life Line Balancing Problem (LBP) at a metalworking company. Based on the concerns of the company's managers, the number of workstations is considered as the main performance criteria for line balancing, while line efficiency and smoothness are considered as the second and the third performance criteria. Two alternative line configurations, i.e., the straight line and the U-shaped line, are considered in the study as a company request. The Grouping Evolution Strategy (GES) algorithm, as one of the novel and powerful method in the literature, is used at a first step, assuming deterministic task times. At a second step, discrete event simulation is used to analyse the impact of stochastic times on the dynamic behaviour of the two line configurations in terms of throughput time and throughput rate. Results of this study are expected to contribute to both research and practice in the context of line balancing.

### ARTICLE INFO

Article history:

Received October 13, 2020

Revised December 25, 2020

Accepted December 28, 2020

Published online January 27, 2021

Keywords:

Line Balancing;

Grouping Evolution Strategy;

Metaheuristic;

Discrete Event Simulation

\*Corresponding author:

Nima Mirzaei

[nimamirzaei@aydin.edu.tr](mailto:nimamirzaei@aydin.edu.tr)

## 1. Introduction

A manufacturing line is a flow-oriented system consisting of a sequence of workstations that performs a repetitive sets of tasks on products [1]. Line balancing may be defined as the decision process of assigning tasks to workstations considering one or more optimization criteria, such as number of workstations and line efficiency [2], [3]. Line balancing is a common problem in practice and can be classified base on the products models being produced (single-model or mix-model), the nature of task times (deterministic or stochastic) and the type of the production flow (straight-line or U-shaped line), between

others [4].

The simple line balancing problem (SLBP), also known as a one-sided or straight line balancing problem, splits the total workload of the product into tasks, and assigns tasks to workstations, so that precedence relations between tasks are not violated. Boysen et al. [5] classifies the SLBP in four different types: type-I, where the objective is to minimize the number of workstations for a given cycle time; type-II, where the objective is to minimize the cycle time for a given number of workstations; type-E, where the objective is to maximize line efficiency and simultaneously adjust both, the number of workstations and cycle time; and type-F, where the objective is to find

a feasible balance for a given cycle time and number of workstations. Problems that generalize or remove assumptions of SLBP make part of the generalized line balancing problem (GLBP). This class of problems contains all extensions that may be relevant in practice, such as U-shaped line, multi-objective, mixed-model products, costs, etc. [6]. For instance, the U-shaped line balancing problem (ULBP) differs from SLBP only by a different interpretation of precedence constraints.

In fact, the straight-line and the U-shaped lines are the most well-known configurations of manufacturing and assembly lines [7]. U-shaped lines have become popular since the introduction of Just-in-Time by Toyota [8]. In U-shaped lines, the product loading and unloading points are set on the same location, i.e., at the end of edge of the U, allowing the operating personnel to work on both sides of the line [9]. Workstations that require working on both sides of the line, are here referred as crossover workstations. In general, if the number of crossover workstations rises, it means that more opportunities for load balancing are created [10]. The U-shaped line has some benefits in comparison with straight-line, such as more flexibility in dealing with demand changes and less space requirements [11], [12], between others [13].

Existing line balancing methods for both, the straight line and the U-shaped line, are generally categorized into exact, heuristic, and metaheuristic [14]. Choosing between them in practice is usually related to the size of the problem [14]. For instance, the Computer Method of Sequencing Operations for Assembly Lines (COMSOAL), proposed by Arcus [15], is a heuristic approach for the assembly line balancing problem. This method can be applied to a variety of problems including small or large size problems because of its heuristic nature [16]. The Ranked Positional Weight (RPW) method is another heuristic method that was used for solving assembly line balancing problems. This method assists to optimize (minimize) the number of workstations, and at the same time, while increasing line efficiency [17]. During recent decades, various studies have been performed on ULBPs, particularly aiming to minimize the number of workstations for a fixed cycle time [9]. Scholl and Klein proposed ULINO (an optimizer for U-shaped lines), which is based on the branch and bound approach [18]. Hwang [19] developed an approach based on a genetic algorithm, and Nilakantan et al. [20] proposed a method based on a particle swarm optimization approach to calculate an automated ULBP. They also pointed out that the

line efficiency of the ULB is higher than that of the straight-line.

Grouping type problems deal with clustering objects in a set of disjoint groups in a way that the union of the groups includes all the objects [21]. In the last decade, evolutionary algorithms have been frequently used for grouping problems' optimization [22]. One of the well-known evolutionary approaches is the Grouping Evolution Strategy (GES) algorithm, which recently has been proposed for crisp grouping problems [18]. This approach is well-matched with the Evolution Strategy (ES) proposed by Rechenberg [19]. The main difference between these two methods is the ES uses Gaussian mutation in the optimization procedure, while GES uses a new and different mutation operation [23], see section 2.3. It has been shown in the literature that GES is one of the most suitable and effective approaches for grouping structured problems such as clustering of fuzzy data [21], Balancing of assembly lines [24], scheduling of batch processing machine, and bin packing problems [22], [23].

This study was triggered by a real-case problem at a metalworking company, requiring a new manufacturing line for one of its most demanded products. Two alternative line configurations, i.e., the straight line and the U-shaped line, are considered in the study as a company request. While a metaheuristic is used to allow obtaining a first approach to line balancing (in the long run), i.e., to determine how tasks should be assigned to workstations, there are temporary fluctuations that require additional control of workloads when dealing with a stochastic environment. Therefore, a discrete event simulation is used to model the release of work to the line at a rate that ensures that the utilization of the bottleneck machine does not exceed and advisable 90% utilization, and to analyse the impact of stochastic task times on the performance of the two line configurations concerning the throughput time and the throughput rate.

To the best of our knowledge existing literature on line balancing did not seek to address the release of work into the production line. Therefore, this study links two streams in the literature, line balancing and job release.

The remainder of this paper is organized as follows: the case study, problem formulation and the solution method used for line balancing are introduced in section 2; then computational results are presented and discussed in section 3. Afterwards, the simulation study carried out is detailed in section 4. Finally, conclusions, limitations and future research work are summarized in section 5.

## 2. Case Study, Problem Formulation and Solution Method

### 2.1 Case Study

The underlying objective of this work is to propose a new manufacturing line for a single product in a metalworking company. Two alternative line configurations, i.e., the straight line and the U-shaped line, are considered in the study as a company request. Unfortunately, when the factory was founded, in 1991, machines and equipment were installed without using a systematic arrangement approach. This resulted in several inefficiencies, including bottleneck operations, production delays and wasting costs. Currently, the company is a prominent producer of engine control switch boxes for the global market. This product has a high demand with a marginal profit for the company. Nevertheless, the product is manufactured in an inefficient layout configuration. Therefore, the company decided to set up a dedicated production line for this high-volume product.

The original layout of the factory, shown in Appendix A, consists of three main buildings. The pro-

duction area is located at the bottom of the layout, and most of the operations are located there, except for painting, assembly and quality control. Different machines such as drills, lathes, etc. have been installed in that area. Therefore, the company has defined a general-purpose area, which has been used mostly for storage, to be used for the new production line.

Due to data protection and privacy, it is not possible to detail the production processes. However, Appendix B depicts the tasks required by the product under study and the precedence relationship between these tasks. Component parts are received from suppliers or produced in the factory at upstream processes. Several operations such as drilling, lathing, sawing, milling, and grinding are performed on these parts involving tasks 1-15. Next, parts are joined to each other by a welding process through subassembly operations (tasks 16-18). After subassembly operations are completed, subassemblies are galvanized and assembled into a final assembly (task 22). Afterwards quality control is performed for every final product (task 23).

Table 1 resumes tasks' information concerning the product under study. The first and the second

**Table 1.** Data set of the case study

| Task Number | Element description | Type of operation | Predecessor   | Task times (min:sec) |       |       |       |
|-------------|---------------------|-------------------|---------------|----------------------|-------|-------|-------|
|             |                     |                   |               | Min                  | Mean  | Max   | Mode  |
| 1           | Drilling            | Manual            | ----          | 5:14                 | 6:11  | 7:00  | 5:45  |
| 2           | Milling             | Manual            | 1             | 2:06                 | 2:24  | 2:33  | 2:20  |
| 3           | Lathing             | Semi-Automated    | 2             | -                    | 3:30  | -     | -     |
| 4           | Drilling            | Manual            | ----          | 4:48                 | 5:40  | 6:07  | 6:00  |
| 5           | Milling             | Manual            | 4             | 2:13                 | 2:24  | 2:35  | 2:22  |
| 6           | Drilling            | Manual            | ----          | 3:01                 | 3:16  | 3:33  | 3:13  |
| 7           | Drilling            | Manual            | ----          | 2:55                 | 3:10  | 3:19  | 3:01  |
| 8           | Drilling            | Manual            | ----          | 3:02                 | 4:03  | 4:07  | 3:30  |
| 9           | Sawing              | Manual            | ----          | 3:02                 | 3:18  | 3:43  | 3:23  |
| 10          | Lathing             | Semi-Automated    | 9             | -                    | 5:34  | -     | -     |
| 11          | Grinding            | Manual            | ----          | 2:01                 | 2:40  | 2:51  | 2:37  |
| 12          | Sawing              | Manual            | ----          | 1:28                 | 2:00  | 2:22  | 1:30  |
| 13          | Drilling            | Manual            | 12            | 16:60                | 17:40 | 17:45 | 17:39 |
| 14          | Lathing             | Semi-Automated    | 13            | -                    | 5:25  | -     | -     |
| 15          | Drilling            | Manual            | ----          | 1:00                 | 1:13  | 1:30  | 1:13  |
| 16          | Welding             | Manual            | 3, 5, 6, 7, 8 | 11:51                | 12:15 | 13:34 | 12:55 |
| 17          | Welding             | Manual            | 10, 11        | 4:14                 | 4:45  | 4:55  | 4:40  |
| 18          | Welding             | Manual            | 14, 15        | 23:00                | 24:50 | 25:24 | 24:40 |
| 19          | Galvanizing         | Fully Automated   | 16            | -                    | 10:00 | -     | -     |
| 20          | Galvanizing         | Fully Automated   | 17            | -                    | 10:00 | -     | -     |
| 21          | Galvanizing         | Fully Automated   | 18            | -                    | 10:00 | -     | -     |
| 22          | Final Assembly      | Manual            | 19, 20, 21    | 12:30                | 13:50 | 15:56 | 13:08 |
| 23          | Quality control     | Manual            | 22            | 15:00                | 16:10 | 20:00 | 17:54 |

column of the table refer to the task number and description, respectively. The third column contains information about the task type (manual, semi-automated, or fully automated). The direct predecessor for each task is in the fourth column, and the last four columns of the table contain the task times concerning the min., mean, max. and mode values, respectively. Times are in min:sec (00:00), in which the first two-digit represent the minute and the last two-digit represent the second.

### 2.2 Problem Formulation

This study focuses on balancing both, a straight line and a U-shaped line, in which a single product is manufactured. Once a deterministic approach is considered here the mean of task times  $t_j$  presented in Table 1 was used for each task  $j$ . Since the largest task mean time is 24:50 min:sec (task 18), the cycle time  $C$  for line balancing purpose was considered to be 30 minutes. Hence, our objective was defined as to minimize the number of workstations given the cycle time, i.e., a line balancing problem of type-I is considered. The (theoretical) minimal number of workstations,  $m^*$ , which is given by the Equation (1), is 6.

$$m^* = \frac{\sum_{j=1}^N t_j}{C} \tag{1}$$

Where  $N$  is the total number of tasks required by the product.

In the line balancing problem under study we follow the literature [25] and considered the following assumptions:

- The precedence relationship between tasks is known beforehand.
- Except for the precedence constraints, there are no other assignment constraints.
- Tasks cannot be split and processed in more than one workstation.
- Only one worker is assigned to each workstation.
- Task times are assumed to be deterministic.
- The sum of all the task times assigned to a workstation cannot exceed the given cycle time.

The LBP that minimizes the number of workstations,  $m$ , given a cycle time  $C$  can be stated as follows:

$$\text{Min. } m = \sum_{i=1}^M \text{Max}_{j=1 \dots N} \{x_{ij}\} \tag{2}$$

Subject to:

$$\sum_{i=1}^M x_{ij} = 1, \forall j = 1, \dots, N \tag{3}$$

$$\sum_{i=1}^M i x_{ik} \leq \sum_{i=1}^M i x_{ij}, \forall j = 1, \dots, N; k \in \text{Pre}(j) \tag{4}$$

$$\sum_{j=1}^N t_j x_{ij} \leq C, \forall i = 1, \dots, M \tag{5}$$

$$x_{ij} = 0, 1, \forall i = 1, \dots, M, j = 1, \dots, N \tag{6}$$

Where  $i$  is the index of the workstations,  $j$  and  $k$  are the indexes of the tasks,  $M$  is the number of workstations and  $N$  is the number of tasks.  $\text{Pre}(j)$  is the set of direct predecessors of task  $j$ , and finally,  $x_{ij}$  is 1, if task  $j$  is assigned to workstation  $i$ , and 0, otherwise.

In this mathematical model, the objective given by equation (2) is to minimize the number of workstations  $m$  at the production line. The first constraint, equation (3), is used to formulate the general feasibility of the problem and states that each task must be assigned to one and only one workstation. Equation (4) states that if a task is assigned to a workstation, then the predecessor of this task must be already assigned to this workstation or to a previous one. The SLBP and ULBP are represented by the same model except for the precedence constraints [26]. In the former, a task is assigned to a workstation after all its predecessors, while in the later a task is assigned after all its predecessors or successors. See Aase et al. [27], [28] to formulate precedence constraints for the U shaped line. Equation (5) denotes that the available time at each workstation should be less than or equal to the cycle time, and finally, equation (6) represents the usual integrity restriction [29].

To evaluate the performance of the obtained solution, the Line Efficiency (E) and the Smoothness Index (SI) determined according to equations (7) and (8), respectively, are used. According to the nature of these indexes, the first one should be maximized and the second one should be minimized.

$$E = \frac{\sum_{i=1}^m T(S_i)}{m \cdot C} * 100 \quad (7)$$

$$SI = \sqrt{\frac{\sum_{i=1}^m (C - T(S_i))^2}{m}} \quad (8)$$

Where  $S_i$  is the set of tasks assigned to workstation  $i$ ,  $m$  is the number of workstations that resulted from the assignment process and  $T(S_i)$  is the total time of the tasks assigned to  $i$ .

## 2.2 Solution Method

Line balancing problems belongs to the class of NP-hard problems [6]. Since there is no exact method to solve these kind of problems in a polynomial CPU-time for large-size problems, only metaheuristic methods have been considered in the study. We selected one of the most effective methods, that recently was proposed by Nejad et al. [30] to solve the SLBP of type-I, which is based on grouping evolution strategy (GES) algorithm.

GES is a kind of evolutionary algorithm that was proposed for grouping problems. Similar to other meta-heuristics, in GES, to escape from trapping the local optima, an operator which is called “mutation” is used. This operation in the GES approach randomly eliminates some of the tasks from their workstations and tries to assign the removed tasks to the remaining workstations, in a way that a better solution, considering the objective function, is achieved. The elimination of tasks from workstations is called “inheritance” and using an effective approach to find a better solution is called the “post assignment”. If  $x_j^t$  is the set of those tasks that have been allocated to workstation  $j$  at iteration  $t$  (the parent solution), and  $y_{ij}^t$  includes all the tasks that are allocated to workstation  $j$  (the offspring) from the parent of  $x_j^t$ , the mutation operator determines both “inheritance” and “post assignment” phases. The following pseudo-code includes the procedure of generating an offspring by using the mutation operator of the GES method. For more details about the mutation operator in GES, the reader is suggested to refer to Kashan et al. [22].

### Phase 1 (inheritance)

For each  $j$  from 1 to the number of workstations in the parent of  $X^t$

The set of  $x_j^t$  contains all the tasks allocated to workstation, say  $S_j$ , in the iteration of  $t$ ;

$$n_{ij}^t \leftarrow \lfloor (1 - \text{Beta}_j(\alpha^t, \beta)) | x_j^t | \rfloor ;$$

$$y_{ij}^t \leftarrow \emptyset ;$$

Find the number of tasks ( $n_{ij}^t$ ) among the tasks of  $x_j^t$ , then, allocate them to  $y_{ij}^t$  ;

End for

### Phase 2 (post assignment)

Assign the removed tasks to workstations up to the time that there will be no task without a workstation and update  $y_{ij}^t$  to find a complete solution of  $Y_i^t$  ;

Where  $\text{Beta}_j(\alpha^t, \beta)$  is a random number generated by using a beta distribution considering two parameters of  $\alpha^t$  and  $\beta$ .

## 3. Computational Results for Line Balancing

This section presents the computational results of the GES method described in the previous section for both, the straight line and the U-shaped line.

Table 2, contains the results concerning the number of workstations ( $m$ ), cycle time ( $C$ ), line efficiency index ( $LE$ ), and smoothness index ( $SI$ ) for the straight-line. Note that the  $S_i$  refers to the set of tasks that were assigned to each workstation  $i$  and  $T(S_i)$  refers to the total time of the tasks assigned to  $i$ .

Results indicate that tasks 7,12 13, 14, and 15 were assigned to the workstation 1 (WS1), tasks 6, 18 were assigned to workstation 2 (WS2), tasks 1, 2, 3, 4 and 5 were assigned to workstation 3 (WS3), tasks 10, 16 and 21 were assigned to workstation 4 (WS4), tasks 11, 17, 19 and 20 were assigned to workstation 5 (WS5), and finally tasks 22 and 23 were assigned to

**Table 2.** Computational results of GES for the straight-line

| GES                                      |                            |  |                                 |                                     |                             |
|--|----------------------------|--|---------------------------------|-------------------------------------|-----------------------------|
| m = 6; C = 30:00; E = 94.61% SI = 113.80 |                            |  |                                 |                                     |                             |
| WS1                                      | WS2                        | WS3                                      | WS4                             | WS5                                 | WS6                         |
| S1: 7, 12, 13, 14, 15<br>T(S1) = 29:28   | S2: 6, 18<br>T(S2) = 28:06 | S3: 1, 2, 3, 4, 5, 8, 9<br>T(S3) = 27:30 | S4: 10, 16, 21<br>T(S4) = 27:49 | S5: 11, 17, 19, 20<br>T(S5) = 29:25 | S6: 22, 23<br>T(S6) = 30:00 |

workstation 6 (WS6). The final result of this method shows that 6 workstations are needed, where C is 30 min, LE is 94.61%, and SI is 113.80.

Table 3 contains the results for the U-shaped line. FS stands for the front-side of the line and BS for back-side. The product enters the U-shaped line at the front-side and exit from the back-side of the line. The remaining parameters in the table are the same as Table 2. It should be mentioned that the underlined ( ) task numbers in Table 3 might be processed in one or another side of the line considering the precedence diagram.

Results of solving the problem by using the GES method illustrates that the task numbers of 1, 2 11, 15 and 23 belong to the workstation 1 (WS1), tasks 9, 12, 13, 14 belongs to workstation 2 and so on. The final result of this method shows that 6 workstations are needed, where C is 28:41 min, LE is 98.95%, and SI is 98.24.

### 4. Simulation Study

In the previous sections, we assume task times to be deterministic. Though this is not the case in the company under study, where task times vary from

cycle to cycle, when human operators are engaged. Therefore, in this section, we apply discrete event simulation (DES) to model the impact of uncertain task times.

#### 4.1 Simulation Models

Solutions obtained by the GES method do not reflect operational variability and randomness at the production line. Therefore, simulation models of both, the straight line and the U-shaped line, have been implemented using SIMIO discrete simulation software to analyse the dynamic behaviour of the production system. The GES meta-heuristic results concerning line configurations are used here as input data for the simulation study. Our models are stochastic, whereby tasks times are random (stochastic) variables. Task times are assumed to follow a triangular distribution, based on data provided by the company concerning the min., mode and max. values (see Table 1), while jobs' inter-arrival times at the line were adjusted to result in an advisable 90% utilization rate at the bottleneck workstation.

Fig. 2 illustrates the developed simulations models. In general, these models can be described as follows. Jobs are created in accordance with exponential

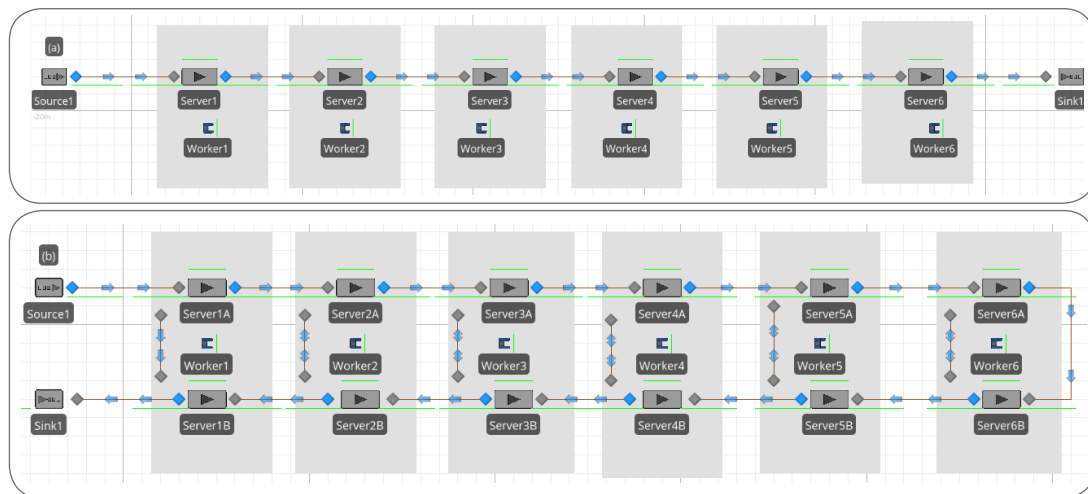


Figure 2. Prediction Speed (obs/sec) for each ML technique

Table 3. Computational results of GES for the U-shaped line

| m= 6; C = 28:41; LE = 98.95%; SI = 98.24 |                  |                   |              |             |                      |                  |
|--|------------------|-------------------|--------------|-------------|----------------------|------------------|
| GES                                      | WS1              | WS2               | WS3          | WS4         | WS5                  | WS6              |
| FS                                       | S1: 1, 2, 11, 15 | S2: 9, 12, 13, 14 | S3: 3, 18    | S4: 8       | S5: 4, 5, 10, 17, 20 | S6: 6, 7, 16, 19 |
| BS                                       | S1: 23           |                   |              | S4: 21, 22  |                      |                  |
| Time                                     | T(S1)= 28:38     | T(S2)= 28:23      | T(S3)= 28:20 | T(S4)=27:53 | T(S5)= 28:23         | T(S6) = 28:41    |

inter arrival times at the source object and release to the assembly line for processing according to the release method. The release time and routing are assigned to the jobs at release. Jobs will then request processing from all servers (workstations) in their routing, in sequence. At each workstation the worker that will perform the operation, the task sequence and the task times are defined. After processing at all stations, statistics concerning e.g., the cycle time and the mean throughput time are collected in the Sink object.

The simulation analysis was developed assuming one worker per workstation, which corresponds to the company practice. This means that a total of six workers are assigned to the line, where the system contains six workstations. It was also assumed that the line is operated under a job release policy that ensures an advisable 90% utilization rate at the bottleneck stations, avoiding in this way overloads that may lead to excessive queues waiting times, and therefore high throughput times.

Simulation analysis is based on three main performance criteria, as follows: the cycle time (C); the mean throughput time (TT) – the mean of the completion date minus the release date across jobs; and utilization rate (Utz) – the percentage of time workstations are occupied. The results are given in the following section.

## 4.2 Simulation Results and Analysis

Results were collected over 10,000 time units (min.) following a warm-up period of 2,000 time units (min.), where each experimental scenario was replicated 100 times. These simulation conditions are in line with those used in previous studies that applied simulation e.g., [31], allowing to obtain stable results, while keeping the simulation run time to a reasonable level. Moreover, to validate our model we compared simulation and meta-heuristic results under deterministic times.

Performance results under deterministic and stochastic times for the different measures were then re-

corded. Table 4 shows these results for both line configurations, the straight line and the U-shaped line.

Simulation results show that cycle times under deterministic task times (i.e., 30:00 min:sec, for straight line and 28:42 min:sec for U-shaped line) are not achievable in practice, i.e., under system variability. That is, to reach an advisable 90% utilization at the bottleneck, cycle times should be set (at least) to 35:00 min:sec, for the straight line and 33:00 min:sec for the U-shaped line. Simulation results confirm that the U-shaped line allows achieving a lower cycle time compared to the straight line. However, results also indicate that the U-shaped line leads to a higher mean of throughput time. For example, under deterministic processing times, the mean of throughput time for the straight line is 170:18 min:sec (i.e., the sum of all task times) while for the U-shaped line is 172:06 min:sec. This is due to worker unavailability, that is once workers operate at both sides of the line (for crossover workstations), in some situations the operator will be needed at the same time at both sides, which leads to delays in the flow of the job. The same happens under stochastic times. However, the probability of occurrence of this kind of situation is higher under stochastic processing times, leading to a higher mean throughput time, as can be observed.

## 4.3 Managerial Implications

The results of this study will be used by the company under study to decide on the line configuration to adopt, i.e., the straight line or the U-shaped line, and how to assign tasks to workstations. Our results indicate that the U-shaped line allows for a higher throughput rate (i.e., about 5.7% lower cycle time). However, this is obtained at the cost of a higher throughput time (i.e., about 10.2%). Higher throughput time means that the delivery times to costumers will also be higher, which may impact the company competitiveness. However, when deciding, the company should also take into account that the U-shaped line contributes to improve visibility and communication skills between operators, while allowing adjusting

**Table 4.** Simulation results for both the Straight line and the U-Shaped line

| Configuration | Time Assumption | C (min:sec) | TT (min:sec) | Utz1 (%) | Utz2 (%) | Utz3 (%) | Utz4 (%) | Utz5 (%) | Utz6 (%) |
|---------------|-----------------|-------------|--------------|----------|----------|----------|----------|----------|----------|
| Straight line | Deterministic   | 30:00       | 170:18       | 98.3     | 93.7     | 91.7     | 92.7     | 91.4     | 100      |
|               | Stochastic      | 35:00       | 170:25       | 82.8     | 78.9     | 76.5     | 81.0     | 77.4     | 90       |
| U-shaped line | Deterministic   | 28:42       | 172:06       | 99.7     | 98.9     | 98.7     | 97.1     | 98.9     | 100      |
|               | Stochastic      | 33:00       | 187:47       | 90       | 84.9     | 84.4     | 83.0     | 85.5     | 88.3     |

the number of workers in the line to react with more flexibility to demand changes.

## 5. Conclusion

In this study, a real-life balancing problem, aiming to minimize the number of workstations given the cycle time was studied. The problem is based on a single-product case of a metalworking manufacturing company. Both the straight line and the U-shape line have been considered as alternative configurations for the production line.

To find the best solution for this real-life case, the GES metaheuristic has been used for line balancing. Additionally, an integrated approach that combines line balancing and discrete event simulation to analyse its dynamic behaviour was used.

GES allows us to find the best solution concerning how to assign tasks to workstations for both, the straight line and the U-shaped line, assuming deterministic processing times. The U-shaped line is better than the straight-line considering line efficiency and smoothness index, while having the same number of workstations. However, simulation results, considering stochastic task times indicate that U-shaped line has higher throughput times, while allowing for a higher throughput rate.

From the results of the simulation, it becomes evident that is 'dangerous' to ignore system variability in line designing. The study also highlights simulation as an important tool to complement line balancing methods, allowing to estimate the impact of stochastic task times.

A limitation of the study is that a single model (product) was considered, while this is justified by the high-volume of the product considered, future research work should extend the study to mixed model lines. Future research work may also analyse the impact of different task times distributions when dealing with stochastic times.

## Acknowledgments

The authors would like to thank the Güral Elektrik Malzemeleri Ticaret ve Sanayi A.Ş (GURAL) for their strong support and advice during the study.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

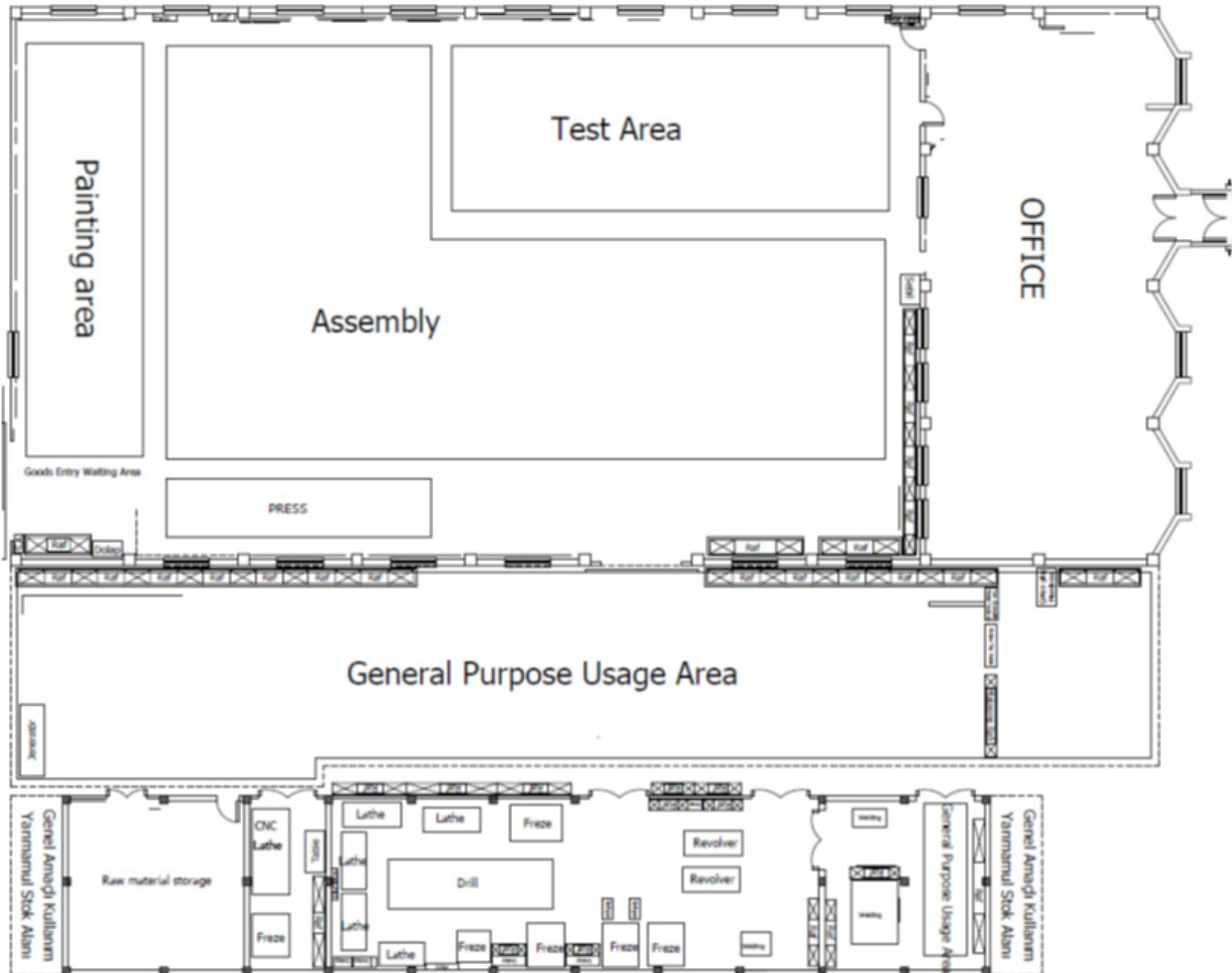
## References

- [1] B. Katalinic, V. E. Pryanichnikov, P. Cesarec, and R. Kettler, "Bionic Assembly System : working modes , control and scheduling," *Int. J. Ind. Eng. Manag.*, vol. 3, no. 3, pp. 121-131, 2012.
- [2] 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.
- [3] S. Digiesi, F. Facchini, G. Mossa, and G. Mummolo, "Minimizing and balancing ergonomic risk of workers of an assembly line by job rotation: A MINLP Model," *Int. J. Ind. Eng. Manag.*, vol. 9, no. 3, pp. 129-138, 2018.
- [4] M. Fathi, A. Nourmohammadi, A. H. C. Ng, and A. Syberfeldt, "An optimization model for balancing assembly lines with stochastic task times and zoning constraints," *IEEE Access*, vol. 7, pp. 32537-32550, 2019.
- [5] 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, 2007.
- [6] A. Scholl and C. Becker, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 694-715, 2006.
- [7] M. G. Nejad and A. H. Kashan, "An Effective Grouping Evolution Strategy Algorithm Enhanced with Heuristic Methods for Assembly Line Balancing Problem," *J. Adv. Manuf. Syst.*, vol. 18, no. 03, pp. 487-509, Sep. 2019.
- [8] I. Kato and A. Smalley, *Toyota Kaizen Methods: Six Steps to Improvement*, 1st ed. CRC Press, 2010.
- [9] M. Li, Q. Tang, Q. Zheng, X. Xia, and C. A. Floudas, "Rules-based heuristic approach for the U-shaped assembly line balancing problem," *Appl. Math. Model.*, vol. 48, pp. 423-439, 2017.
- [10] M. Glonegger and G. Reinhart, "Planning of synchronized assembly lines taking into consideration human performance fluctuations," *Prod. Eng.*, vol. 9, no. 2, pp. 277-287, 2015.
- [11] G. Reinhart, J. Werner, and F. Lange, "Robot based system for the automation of flow assembly lines," *Prod. Eng.*, vol. 3, no. 1, pp. 121-126, Mar. 2009.
- [12] M. Şahin and T. Kellegöz, "Increasing production rate in U-type assembly lines with sequence-dependent set-up times," *Eng. Optim.*, vol. 49, no. 8, pp. 1401-1419, 2017.
- [13] A. C. Alves, "U-Shaped Cells Operating Modes : a Review and a Hands-on Simulation Comparison," *Int. J. Ind. Eng. Manag.*, vol. 9, no. 2, pp. 87-97, 2018.
- [14] M. R. Abdullah Make, M. F. F. Mohd Fadzil, and M. M. Razali, "A review of two-sided assembly line balancing problem," *Int. J. Adv. Manuf. Technol.*, vol. 89, no. 5-8, pp. 1743-1763, 2017.
- [15] A. L. ARCUS, "COMSOAL: a computer method of sequencing operations for assembly lines.," *Int. J. Prod. Res.*, no. 4, pp. 259-277, 1966.
- [16] G. W. Depuy and G. E. Whitehouse, "Applying the COMSOAL computer heuristic to the constrained resource allocation problem," vol. 38, 2000.
- [17] W. B. Helgeson and D. P. Birnie, "Assembly Line Balancing Using the Ranked Positional Weighting Technique," *J. Ind. Eng.*, no. 12, pp. 394-398, 1961.
- [18] A. Scholl and R. Klein, "ULINO: Optimally balancing U-shaped JIT assembly lines," *Int. J. Prod. Res.*, vol. 37, no. 4, pp. 721-736, Mar. 1999.
- [19] R. K. Hwang, H. Katayama, and M. Gen, "U-shaped assembly line balancing problem with genetic algorithm," *Int. J. Prod. Res.*, vol. 46, no. 16, pp. 4637-4649, Aug. 2008.
- [20] M. J. Nilakantan, S. G. Ponnambalam, and N. Jawahar,



- “Design of energy efficient RAL system using evolutionary algorithms,” *Eng. Comput.*, vol. 33, no. 2, pp. 580–602, Apr. 2016.
- [21] A. Husseinzadeh Kashan, B. Rezaee, and S. Karimiyan, “An efficient approach for unsupervised fuzzy clustering based on grouping evolution strategies,” *Pattern Recognit.*, vol. 46, no. 5, pp. 1240–1254, May 2013.
- [22] A. H. Kashan, A. A. Akbari, and B. Ostadi, “Grouping evolution strategies: An effective approach for grouping problems,” *Appl. Math. Model.*, vol. 39, no. 9, pp. 2703–2720, 2015.
- [23] A. H. Kashan, M. Jenabi, and M. H. Kashan, “A New Solution Approach for Grouping Problems Based on Evolution Strategies,” in *2009 International Conference of Soft Computing and Pattern Recognition, 2009*, pp. 88–93.
- [24] M. Ghadirinejad, A. Husseinzadeh Kashan, and F. Rismanchian, “A new competitive method for solving assembly line balancing problem,” in *1st international conference on new directions in business, management, finance and economics.*, 2013, p. 73.
- [25] B. Rekiek and A. Delchambre, *Assembly Line Design: The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms*. London: Springer-Verlag, 2006.
- [26] A. Fattahi and M. Turkay, “On the MILP model for the U-shaped assembly line balancing problems,” *Eur. J. Oper. Res.*, vol. 242, no. 1, pp. 343–346, Apr. 2015.
- [27] G. R. Aase, M. J. Schniederjans, and J. R. Olson, “U-OPT: An analysis of exact U-shaped line balancing procedures,” *Int. J. Prod. Res.*, vol. 41, no. 17, pp. 4185–4210, Jan. 2003.
- [28] G. R. Aase, J. R. Olson, and M. J. Schniederjans, “U-shaped assembly line layouts and their impact on labor productivity: An experimental study,” *Eur. J. Oper. Res.*, vol. 156, no. 3, pp. 698–711, Aug. 2004.
- [29] M. Gen, R. Cheng, and L. Lin, *Network Models and Optimization*. London: Springer London, 2008.
- [30] M. Ghadiri Nejad, A. Husseinzadeh Kashan, and S. M. Shavarani, “A novel competitive hybrid approach based on grouping evolution strategy algorithm for solving U-shaped assembly line balancing problems,” *Prod. Eng.*, vol. 12, no. 5, pp. 555–566, Oct. 2018.
- [31] N. O. Fernandes, M. Thürer, T. M. Pinho, P. Torres, and S. Carmo-Silva, “Workload control and optimised order release: an assessment by simulation,” *Int. J. Prod. Res.*, pp. 1–14, Jun. 2019.

## Appendix A: Current layout of the industrial unit



## Appendix B: The precedence relationship chart of the case study

